



**UNIVERSIDAD NACIONAL DE INGENIERÍA**  
**Facultad de Electrotecnia y Computación**

**SISTEMA DE GESTIÓN DE CONTENIDO PARA PORTALES CAUTIVOS Y  
OPTIMIZACIÓN DEL PANEL DE ADMINISTRACIÓN DE CONEXIONES WIFI  
PARA LA EMPRESA WIMIX SOLUTIONS AND SERVICES**



Trabajo monográfico

Presentado por:

Br. María Natalia Flores Pavón – 2012-41101

Br. Billi Alexander Ramírez Obregón – 2012-41690

Tutor:

MSc. Ing. Lizette Duarte Mora

Mayo 2019

Managua, Nicaragua





## **DEDICATORIA**

*Dedicamos este trabajo monográfico a Dios, por permitirnos la gracia de la vida, y de llegar a esta etapa especial en nuestras vidas, por brindarnos sabiduría para seguir adelante en nuestro crecimiento personal y profesional.*

*A nuestras madres por su apoyo incondicional, por su guía desde nuestros primeros pasos hasta la actualidad, por sus consejos y sacrificios durante todos estos años.*

*A todos nuestros compañeros y amigos que sin esperar nada a cambio, nos compartieron conocimientos, alegrías y tristezas; y a todas esas personas que estuvieron a nuestro lado apoyándonos durante estos años de estudios.*

*Muchas gracias a cada uno.*



## **AGRADECIMIENTO**

*Agradecemos a Dios por todas las bendiciones que nos ha brindado, nuestras familia y amigos, que siempre han estado a nuestro lado. Gracias por cada día de nuestras vidas y por permitirnos seguir adelante, por las decisiones que hemos tomado que han sido lección y aprendizaje en nuestra vida profesional.*

*A nuestras mamás, que nos ha brindado su apoyo incondicional, por los sacrificios hechos para que nuestros estudios fueran concluidos, por sus consejos, regaños y cariño. Sin ellas, no estaríamos hoy por hoy, culminando una etapa más de nuestras vidas. Muchas gracias siempre.*

*A nuestra tutora, Msc. Lizette Duarte. Muchas gracias por confiar en nosotros, por guiarnos y aconsejarnos a lo largo del desarrollo de este trabajo monográfico. Su apoyo fue muy importante para continuar el mismo.*



## RESUMEN DEL TEMA

El presente proyecto monográfico titulado “Sistema De Gestión De Contenido Para Portales Cautivos Y Optimización Del Panel De Administración De Conexiones Wifi Para La Empresa Wimix Solutions And Services” contempla el desarrollo de un sistema que potencialice la marca, a la empresa y a sus clientes.

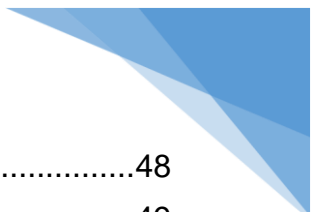
Wimix Solutions and Services actualmente cuenta con una plataforma para visualizar las conexiones y los usuarios registrados desde un portal cautivo llamado WimixDash. Sin embargo, dicha plataforma no está creada ni optimizada para ser reutilizada con diferentes clientes. Con el nuevo sistema, se brindará una forma sencilla para que los clientes de Wimix puedan realizar cambios en sus plataformas, de esta manera se reduce la dependencia entre ellos y se garantiza la agilización de dicho proceso.

A lo largo del documento, leerá una descripción de la empresa y sus procesos actuales, aspectos que fueron punto de partida para determinar la vitalidad del proyecto.

Para el desarrollo del sistema, se utilizó la metodología XP (Xtreme Programming), la cual permite una retroalimentación continua y en cortos plazos entre el equipo del proyecto y el cliente. Esta rápida comunicación concede cambios incrementales, simplificando alcanzar los objetivos y reduciendo los costos del proyecto.

## TABLA DE CONTENIDO

|       |   |    |
|-------|---|----|
| I.    | INTRODUCCIÓN .....  | 2  |
| II.   | OBJETIVOS .....   | 3  |
| 2.1.  | Objetivo General .....  | 3  |
| 2.2.  | Objetivos Específicos .....                                       | 3  |
| III.  | JUSTIFICACIÓN .....   | 4  |
| IV.   | MARCO TEÓRICO.....  | 7  |
| 4.1.  | CMS (Content Management System).....                              | 7  |
| 4.1.1 | Definición .....  | 7  |
| 4.1.2 | Características .....   | 8  |
| 4.1.3 | Clasificación .....   | 9  |
| 4.2.  | Arquitectura de Software: Cliente - Servidor .....                | 10 |
| 4.3.  | Herramientas de desarrollo .....                                  | 11 |
| 4.3.1 | Tecnologías Frontend .....  | 11 |
| 4.3.2 | Tecnologías Backend.....  | 13 |
| 4.4.  | Seguridad .....   | 24 |
| 4.5.  | Estándares web.....   | 27 |
| 4.6.  | Redes .....   | 29 |
| 4.6.1 | RADIUS .....  | 29 |
| 4.6.2 | Servidor RADIUS .....   | 29 |
| 4.6.3 | Portal Cautivo.....   | 30 |
| 4.7.  | <i>Extreme Programming</i> (XP) como metodología de desarrollo .. | 31 |
| V.    | ESTUDIO DE FACTIBILIDAD .....                                     | 36 |
| 5.1.  | Aspectos Técnicos .....   | 36 |
| 5.1.1 | Inventario de recursos de hardware y software.....                | 36 |
| 5.1.2 | Estrategia del Hardware, Software y Servicios .....               | 38 |
| 5.2.  | Aspectos Económicos .....   | 41 |
| 5.2.1 | Costo de Desarrollo .....   | 41 |
| 5.3.  | Aspectos Operativos .....   | 44 |
| 5.3.2 | Cronograma .....  | 45 |
| 5.4.  | Aspectos Legales .....  | 46 |
| VI.   | APLICACIÓN DE LA METODOLOGÍA XP .....                             | 48 |



|       |   |     |
|-------|---|-----|
| 6.1.  | Fase de Exploración.....                  | 48  |
| 6.1.2 | Modelo del Negocio .....                  | 49  |
| 6.1.4 | Requerimientos del sistema .....          | 50  |
| 6.1.5 | Historias Usuarios .....                  | 51  |
| 6.2.  | Fase de Planificación .....               | 57  |
| 6.2.1 | Cronograma de tareas .....                | 58  |
| 6.3.  | Fase de Diseño .....                      | 62  |
| 6.3.1 | Diagramas de actividades .....            | 62  |
| 6.3.2 | Diagramas de secuencia.....               | 66  |
| 6.3.3 | Diagramas de navegación.....              | 71  |
| 6.3.4 | Diagrama lógico de la base de datos ..... | 74  |
| 6.3.5 | Diseño del Interfaz de Usuario .....      | 82  |
| 6.4.  | Fase de Codificación .....                | 87  |
| 6.4.1 | Arquitectura de software .....            | 87  |
| 6.4.2 | Estructura del sistema.....               | 89  |
| 6.5.  | Fase de Prueba .....                      | 94  |
| 6.5.1 | Pruebas de funcionalidad.....             | 95  |
| 6.5.2 | Pruebas de seguridad .....                | 101 |
| 6.6.  | Fase de producción .....                  | 103 |
| VII.  | CONCLUSIONES Y RECOMENDACIONES .....      | 105 |
| VIII. | REFERENCIAS.....                          | 108 |

## IMÁGENES

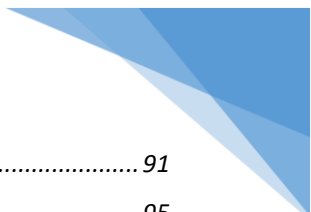
|   |           |
|---|-----------|
| <i>Imagen 1 Arquitectura Cliente – Servidor.....</i>                                      | <i>10</i> |
| <i>Imagen 2 Tipos de colecciones en una base de datos no relacional.....</i>              | <i>15</i> |
| <i>Imagen 3 Mapeo de objetos entre Node y MongoDB .....</i>                               | <i>17</i> |
| <i>Imagen 4 Fuente: (Castro, 2005) .....</i>  | <i>30</i> |
| <i>Imagen 5 Fases de la metodología X (Pressman, 2010) Traducción propia .....</i>        | <i>32</i> |
| <i>Imagen 6 Cronograma.....</i>   | <i>45</i> |
| <i>Imagen 7 Modelo del negocio .....</i>  | <i>49</i> |
| <i>Imagen 8 Cronograma de tareas. Iteración A .....</i>                                   | <i>58</i> |
| <i>Imagen 9 Cronograma de tareas. Iteración B .....</i>                                   | <i>59</i> |
| <i>Imagen 10 Cronograma de tareas. Iteración C.....</i>                                   | <i>60</i> |
| <i>Imagen 11 Diagrama de actividades - Autenticación.....</i>                             | <i>62</i> |
| <i>Imagen 12 Diagrama de actividades - Gestión de formulario de registro.....</i>         | <i>63</i> |
| <i>Imagen 13 Diagrama de actividades - Gestión de formulario de encuestas .....</i>       | <i>64</i> |
| <i>Imagen 14 Diagrama de actividades – Conexiones Recientes.....</i>                      | <i>65</i> |
| <i>Imagen 15 Diagrama de actividades - Administración de promociones .....</i>            | <i>65</i> |
| <i>Imagen 16 Diagrama de actividades - Visualización de dispositivos registrados.....</i> | <i>66</i> |
| <i>Imagen 17 Diagrama de actividades – Panel principal / Dashboard .....</i>              | <i>66</i> |
| <i>Imagen 18 Diagrama de secuencia - Autenticación .....</i>                              | <i>67</i> |
| <i>Imagen 19 Diagrama de secuencia - Gestión de formulario de registro .....</i>          | <i>68</i> |
| <i>Imagen 20 Diagrama de secuencia - Gestión de formulario de encuestas.....</i>          | <i>69</i> |
| <i>Imagen 21 Diagrama de secuencia – Conexiones Recientes .....</i>                       | <i>70</i> |
| <i>Imagen 22 Diagrama de navegación para usuario administrador.....</i>                   | <i>71</i> |
| <i>Imagen 23 Diagrama de navegación para usuario establecimiento cliente .....</i>        | <i>72</i> |
| <i>Imagen 24 Diagrama de navegación para usuario general .....</i>                        | <i>73</i> |
| <i>Imagen 25 Nomenclatura para la Base de Datos.....</i>                                  | <i>74</i> |
| <i>Imagen 26 Colecciones de la base de datos I .....</i>                                  | <i>74</i> |
| <i>Imagen 27 Colecciones de la base de datos II .....</i>                                 | <i>75</i> |
| <i>Imagen 28 Colecciones de la base de datos III .....</i>                                | <i>75</i> |
| <i>Imagen 29 Colecciones de la base de datos IV.....</i>                                  | <i>76</i> |
| <i>Imagen 30 Colecciones de la base de datos V.....</i>                                   | <i>76</i> |
| <i>Imagen 31 Colecciones de la base de datos VI.....</i>                                  | <i>77</i> |
| <i>Imagen 32 Colecciones de la base de datos VII .....</i>                                | <i>77</i> |
| <i>Imagen 33 Relaciones por referencia entre colecciones I.....</i>                       | <i>78</i> |
| <i>Imagen 34 Relaciones por referencia entre colecciones II.....</i>                      | <i>79</i> |



|   |           |
|---|-----------|
| <i>Imagen 35 Relaciones por referencias entre colecciones III .....</i> | <i>80</i> |
| <i>Imagen 36 Relaciones por referencia entre relaciones IV .....</i>    | <i>81</i> |
| <i>Imagen 37 Relaciones por referencia entre colecciones V.....</i>     | <i>82</i> |
| <i>Imagen 38 Diseño del sistema - Estructura principal .....</i>        | <i>83</i> |
| <i>Imagen 39 Diseño del sistema - Menú lateral .....</i>                | <i>84</i> |
| <i>Imagen 40 Diseño del sistema – Notificaciones .....</i>              | <i>84</i> |
| <i>Imagen 41 Diseño del sistema - Módulos.....</i>                      | <i>84</i> |
| <i>Imagen 42 Diseño del sistema - Modales .....</i>                     | <i>85</i> |
| <i>Imagen 43 Arquitectura del software.....</i>                         | <i>88</i> |
| <i>Imagen 44 Carpeta de archivos - Server.....</i>                      | <i>89</i> |
| <i>Imagen 45 Carpeta de archivos - Client.....</i>                      | <i>90</i> |
| <i>Imagen 46 Estructura del sistema .....</i>                           | <i>92</i> |

## **TABLAS**

|   |           |
|---|-----------|
| <i>Tabla 1 Inventario de recursos tecnológicos de Wimix Solutions and Services .....</i>    | <i>36</i> |
| <i>Tabla 2 Requerimientos tecnológicos mínimos para la implementación del sistema .....</i> | <i>37</i> |
| <i>Tabla 3 Estrategia de Hardware .....</i>   | <i>38</i> |
| <i>Tabla 4 Estrategia de Software / Servicios .....</i>                                     | <i>39</i> |
| <i>Tabla 5 Equipo de desarrollo .....</i>   | <i>41</i> |
| <i>Tabla 6 Costos de desarrollo.....</i>  | <i>41</i> |
| <i>Tabla 7 Costos de Hardware.....</i>  | <i>42</i> |
| <i>Tabla 8 Costos de Software/servicios.....</i>  | <i>43</i> |
| <i>Tabla 9 Costos totales .....</i>   | <i>44</i> |
| <i>Tabla 10 Historia de Usuario – Gestión de formulario de registro .....</i>               | <i>52</i> |
| <i>Tabla 11 Historia de usuario - Gestión de formulario de encuestas.....</i>               | <i>52</i> |
| <i>Tabla 12 Historia de usuario - Configuraciones generales del sistema .....</i>           | <i>53</i> |
| <i>Tabla 13 Historia de usuario - Autenticación y roles de usuario .....</i>                | <i>53</i> |
| <i>Tabla 14 Historia de usuario - Conexiones recientes.....</i>                             | <i>53</i> |
| <i>Tabla 15 Historia de usuario - Administración de promociones.....</i>                    | <i>54</i> |
| <i>Tabla 16 Historia de usuario - Visualización de usuarios registrados.....</i>            | <i>54</i> |
| <i>Tabla 17 Historia de usuario - Panel principal / Dashboard .....</i>                     | <i>54</i> |
| <i>Tabla 18 Historia de usuario - Historial de conexiones.....</i>                          | <i>55</i> |
| <i>Tabla 19 Historia de usuario - Bitácora .....</i>  | <i>55</i> |
| <i>Tabla 20 Historia de usuario - Autenticidad de correos.....</i>                          | <i>55</i> |
| <i>Tabla 21 Establecimiento de prioridades de las HS.....</i>                               | <i>57</i> |
| <i>Tabla 22 Carpeta de archivos - Server .....</i>  | <i>90</i> |



|   |            |
|---|------------|
| <i>Tabla 23 Carpeta de archivos – Client .....</i>  | <i>91</i>  |
| <i>Tabla 24 Plantilla de pruebas .....</i>  | <i>95</i>  |
| <i>Tabla 25 Prueba - Configurar y guardar formulario de registro.....</i>                               | <i>95</i>  |
| <i>Tabla 26 Configurar y guardar formulario de registro .....</i>                                       | <i>96</i>  |
| <i>Tabla 27 Prueba - Configurar y guardar encuestas NPS.....</i>  | <i>97</i>  |
| <i>Tabla 28 Configurar y guardar encuestas personalizadas.....</i>                                      | <i>98</i>  |
| <i>Tabla 29 Prueba - Verificar las conexiones recientes .....</i>                                       | <i>98</i>  |
| <i>Tabla 30 Prueba - Guardar configuración de visualización de promociones.....</i>                     | <i>98</i>  |
| <i>Tabla 31 Prueba - Exportar archivo de usuarios registrados.....</i>                                  | <i>99</i>  |
| <i>Tabla 32 Prueba - Configuración de orden de las secciones pertenecientes al portal cautivo .....</i> | <i>99</i>  |
| <i>Tabla 33 Prueba - Ingresar al sistema – Autenticación .....</i>                                      | <i>101</i> |
| <i>Tabla 34 Prueba - Ingresar al Sistema - Autenticación II.....</i>                                    | <i>101</i> |
| <i>Tabla 35 Acta de entrega .....</i>   | <i>103</i> |



# CAPITULO I:

## Descripción General

- Introducción
- Objetivos
- Justificación

## **I. INTRODUCCIÓN**

Wimix Solutions and Services<sup>1</sup>, es una empresa de tecnología que se dedica a proveer soluciones tecnológicas de calidad, integrando tecnologías sostenibles y rentables.

Entre las soluciones tecnológicas que provee Wimix está una plataforma de publicidad y mercadeo inteligente que se utiliza para crear opciones de publicidad y mercadeo aprovechando la conexión wifi de los establecimientos, proporcionando así un método de promoción de marca, negocio o producto. Para ello, sería de gran conveniencia contar con un CMS (Content Management System) ajustado a las necesidades de los establecimientos clientes.

Los CMS proporcionan autonomía al crear y administrar contenido, principalmente en la web. Las características de este tipo de sistemas son la inclusión de nuevas funcionalidades en la web y mantenimiento de grandes cantidades de páginas, entre otras más.

Wimix también ofrece un servicio de hotspot para la configuración de conexión wifi y bloqueo de usuarios, entre otras funciones. Este servicio es vital en el flujo del modelo del negocio de Wimix, ya que por medio de éste se logra redirigir a los usuarios a portales cautivos. Para esto será de gran utilidad un CMS que logre integrar el servicio de configuración de conexión wifi y la administración de contenido de portales cautivos de los establecimientos clientes.

El interés de esta propuesta monográfica es desarrollar un CMS para Wimix, con el fin de facilitar a sus establecimientos clientes la gestión de contenido de forma independiente y rápida, promover su marca de acuerdo a sus necesidades empresariales, por lo que puede incluir formularios, promociones y encuestas, entre otras características.

---

<sup>1</sup> En adelante referido como Wimix

## **II. OBJETIVOS**

### **2.1. Objetivo General**

- Desarrollar un CMS para portales cautivos y optimización del panel de administración de conexiones Wifi de los clientes de la empresa Wimix Solutions and Services.

### **2.2. Objetivos Específicos**

- Analizar la situación actual de la empresa con el fin de identificar los requerimientos precisos del CMS.
- Optimizar el diseño lógico del sistema por medio de componentes reutilizables para definir patrones de solución a nuevos requerimientos del usuario.
- Codificar los componentes del sistema bajo plataformas de desarrollo web.
- Realizar pruebas del sistema para verificar su correcto funcionamiento.
- Implantar el CMS en la empresa Wimix para que sea accedido por sus clientes.

### **III. JUSTIFICACIÓN**

La autonomía en plataformas web desarrolladas por terceros es un requerimiento primario que las empresas toman en consideración al momento de hacer uso de este tipo de servicios, ya que el cliente desea tener un control total o parcial de la plataforma y su interacción con los usuarios finales, eliminando así, procesos que involucren al personal de la empresa que proporciona la plataforma. Ese es el caso de los CMS, cuya utilidad es permitir que puedan crear contenido web y así gestionarlo, publicarlo y presentarlo, sin necesidad de involucrar al equipo de desarrollo o administración de la plataforma.

DASH WIMIX, una plataforma creada por Wimix para la visualización de estadísticas de conexiones a la red Wifi de un establecimiento, no fue hecha para aceptar cambios, por lo que incorporar nuevos servicios para ofrecer a los clientes ha sumado cierta complejidad y sobretodo duplicidad del trabajo y atrasos en el desarrollo. Casos como las solicitudes de encuestas y soluciones brindadas al cliente corporativo, se ha convertido en una necesidad de optimización por medio del rediseño de componentes reutilizables para brindar estas funcionalidades a otros clientes que demanden este tipo de cambios.

En cuanto a la administración de conexiones, esta no responde de manera eficaz al reporte de conexiones a consecuencia de que, para evitar múltiples registros de conexión de un mismo cliente en un período de tiempo corto, se simulan sesiones. Es decir, el cliente pudo desconectarse de la plataforma por diversos motivos, esto hace que al conectarse otra vez se agregue esa conexión al registro.

El CMS que se desea desarrollar será capaz de incorporar estos nuevos servicios, brindando independencia a los clientes, puesto que facilitará la personalización de diferentes aspectos del portal, el cuál puede incluir nuevos formularios, promociones, anuncios, encuestas, etc. Será un gestor de contenido de tipo específico ya que es totalmente personalizado de acuerdo a las peticiones brindadas por Wimix. Los establecimientos clientes podrán aprovechar su

conexión a internet para promover y potenciar su negocio, marca o producto, asimismo Wimix, como empresa que brinda la plataforma, alcanzará un mayor segmento de mercado.

Las contribuciones más significativas de este trabajo serán:

- Nuevas funciones que permitan visualizar y controlar las conexiones de internet de los usuarios finales.
- Diseño de componentes reutilizables para la implementación de nuevas características en el sistema.
- Una plataforma adaptable a las necesidades de los establecimientos clientes de Wimix.
- Una plataforma que permita a los establecimientos clientes gestionar el contenido de los portales cautivos de manera personalizada para sus usuarios finales.



# CAPITULO II:

## Marco Teórico

- Aspectos generales
- Tecnologías Frontend
- Tecnologías Backend
- Seguridad
- Redes
- Metodología



## IV. MARCO TEÓRICO

### 4.1. CMS (Content Management System)

#### 4.1.1 Definición

Para definir el término CMS se analizarán las definiciones de varios autores.

(Viciano Pérez, 2014) nos menciona su acercamiento a una definición de un CMS:

*“... un sistema de gestión de contenidos (Content Management System en inglés, abreviado CMS) es un programa que permite crear una estructura de soporte (framework) para la creación y administración de contenidos por parte de los participantes principalmente en páginas web.” (2011, s.p.).*

(Michelinakis, 2004) señala que:

*“Los sistemas de gestión de contenidos no son solo un producto o una tecnología. Un SGC se define como un término genérico que se refiere a una amplia gama de procesos que apoyan la ‘próxima generación’ de sitios web de mediana a gran escala. La gestión de contenidos es un proceso que se ocupa de la creación, almacenamiento, modificación, recuperación y visualización de los datos o contenidos” (abstracto).*

Otra definición es la proporcionada por (Jiménez & Moreo, 2006) quienes simplifican el término en:

*“... la gestión de contenido es el proceso mediante el que se hace casar lo que se tiene con lo que los usuarios desean. ... la gestión de contenido puede parecer una forma de crear grandes sitios Web, pero si miramos con más atención, es de hecho un proceso global para la recolección, gestión y publicación de contenido para cualquier formato de salida” (p. 19).*

Podemos decir que los CMS pueden definirse como herramientas estructuradas que permiten la creación, almacenamiento, modificación, recuperación y visualización de datos y contenidos a distintos niveles de gestión y acceso.

#### 4.1.2 Características

Los CMS pueden tener características variables dependiendo de lo que se quiera implementar en el sistema. Sin embargo, deben cumplir con una serie de rasgos que los hace llamarse así, se listan a continuación:

- **Intuitivo:** Debe tener una interfaz gráfica de usuario (GUI) sin opciones complicadas y ofrecer simplicidad en la interfaz de administración (Reyes, 2009). Podrá utilizarlo un usuario sin conocimientos de programación. Sin embargo, se necesitan conocimientos básicos de informática si el sitio será grande.
- **Flexibilidad y fácil personalización:** El CMS permite personalizar tu propio diseño sin que el usuario tenga restricciones (Reyes, 2009). Debe poder adaptarse a las necesidades del negocio (Domínguez & Urra González, 2006).
- **Accesibilidad:** Deben poseer la capacidad de poder acceder a ellos desde cualquier computadora con red, sin necesidad de instalar programas adicionales (Domínguez & Urra González, 2006).
- **Extensible:** Un buen CMS permite incorporar complementos para extender el sitio web según las necesidades del usuario (Reyes, 2009).
- **Seguridad:** Autenticación de usuarios y comprobación de la información circulante (Domínguez & Urra González, 2006).
- **Alto rendimiento y velocidad:** Un CMS debe ser capaz de que las páginas creadas con el sean rápidas al cargar, de otra manera los usuarios abandonarán el sitio (Reyes, 2009).
- **Énfasis en buenas prácticas y estándares:** Debe promover el uso de estándares para asegurar la compatibilidad entre navegadores, un código limpio y puro y de fácil mantenimiento (Reyes, 2009).

### 4.1.3 Clasificación

Existen diferentes maneras de clasificarlos, por el tipo de contenido que gestionan, por las características del servidor, por funcionalidad, por la licencia de software a la que está sujeta.

(Casado Martínez & Córcoles, 2008) clasifica los CMS según el tipo de contenido que se gestionan, los CMS pueden ser:

- Genéricos: Pensados para ofrecer servicios de diferentes tipos y se pueden usar para elaborar blogs, portales, revistas, tiendas. Entre estos tenemos
- como ejemplos: Plone, OpenCMS, MySpace, TYPO3, Mambo, Joomla o Drupal.
- Blogs: Son los especializados para la gestión de diarios personales. Son CMS de blogs: WordPress, B2Evolution, Movable Type, Blogger.
- Wikis: Son un tipo de CMS pensados para que varios usuarios creen o editen un documento o conjuntos de documentos de manera comunitaria. Se mantiene un registro de las modificaciones hechas por cada usuario. Ejemplos de este tipo de CMS son: MediaWiki (el utilizado por Wikipedia), TipiWiki, PikiWiki o PmWiki.
- Foros: Estos permiten la gestión de foros de discusión. Son conocidos: phpBB, PunBB, MyBB, SMF.
- Galería de imágenes: Permiten subir y gestionar imágenes, crear álbumes, enviar fotos a imprimir. Ejemplos de este tipo de CMS son: Gallery, Flickr, PicasaWeb, Coppermine.
- Comercio electrónico: Son CMS especializados para la compra-venta de productos. Algunos ejemplos son: osCommerce, Magent, PrestaShop.
- Específico: Creados para un tratado específico en una empresa o tema especial (Berberá, 2014).

A su vez, se pueden clasificar según el tipo de licencia de software: si es de **código abierto** o **software propietario** (Berberá, 2014).

## 4.2. Arquitectura de Software: Cliente - Servidor

El modelo Cliente/Servidor es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Las aplicaciones Clientes realizan peticiones a una o varias aplicaciones Servidores, que deben encontrarse en ejecución para atender dichas demandas.

El modelo Cliente/Servidor permite diversificar el trabajo que realiza cada aplicación, de forma que los Clientes no se sobrecarguen, cosa que ocurriría si ellos mismos desempeñan las funciones que le son proporcionadas de forma directa y transparente. En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema. Tanto el Cliente como el Servidor son entidades abstractas que pueden residir en la misma máquina o en máquinas diferentes.

Una disposición muy común son los sistemas multicapa en los que el servidor se descompone en diferentes programas que pueden ser ejecutados por diferentes computadoras aumentando así el grado de distribución del sistema.

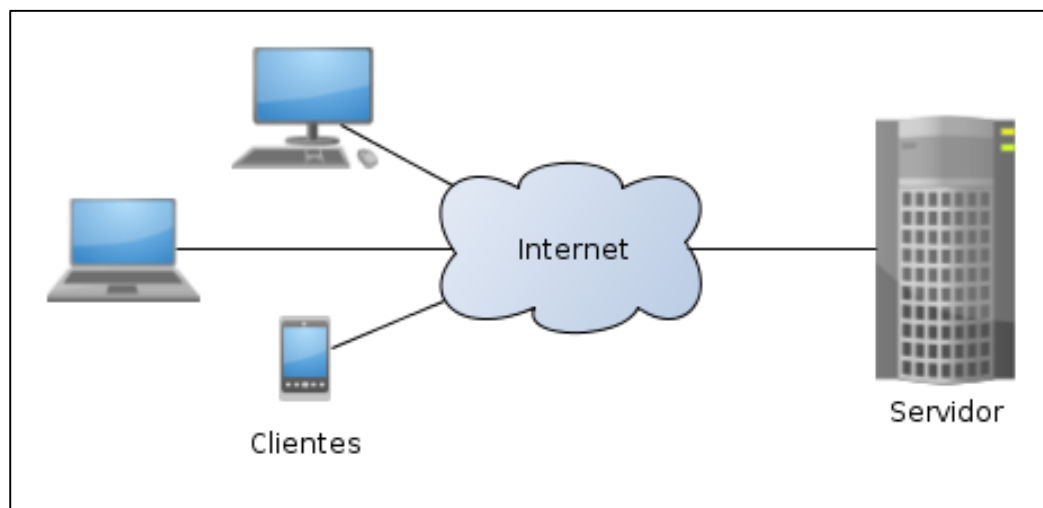


Imagen 1 Arquitectura Cliente – Servidor

## 4.3. Herramientas de desarrollo

### 4.3.1 Tecnologías Frontend

*“Frontend en diseño de software y desarrollo web hace referencia a la visualización del usuario navegante o, dicho de otra manera, es la parte que interactúa con los usuarios. De una manera práctica, el front-end sería todos esos formularios que rellenamos en las webs, las aplicaciones que manejamos y todas esas cosas que tengan que realizar los usuarios.”* (Culturación, s.f.)

**HTML** significa Lenguaje de Marcado para Hipertextos (por sus siglas en inglés, HyperText Markup Language). Es el elemento de construcción más básico de una página web y se usa para crear y representar visualmente una página web. Determina el contenido de la página web, pero no su funcionalidad (Mozilla Developer Network, 2017).

Otras tecnologías distintas de HTML son usadas generalmente para describir la apariencia/presentación de una página web (CSS) o su funcionalidad (Javascript).

**CSS** (Cascading Style Sheets), que en español significa Hojas de Estilo en Cascada es un lenguaje es utilizado para describir la presentación de documentos HTML o XML, incluyendo lenguajes como XHTML O SVG. CSS describe como debe ser renderizado el elemento estructurado en pantalla, en papel, hablado o en otros medios (Mozilla Developer Network, 2017).

**Javascript**, abreviado como JS, según (Mozilla Developer Network, 2017) es un lenguaje ligero e interpretado, orientado a objeto con funciones de primera clase. Es más conocido como el lenguaje de script para páginas web, pero también usado en muchos entornos sin navegador, tales como Node.js o Apache CouchDB.

Javascript es multiparadigma, basado en prototipos, dinámico, soporta estilos de programación funcional, orientada a objetos e imperativa.

Existen muchas librerías basadas en Javascript, algunas de ellas son:

- JQuery: Es una librería de Javascript que se encarga de simplificar la manipulación del DOM, llamadas Ajax, y manejadores de eventos.
- Ajax: JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications).
- React.js: Librería de Javascript de código abierto, ofrece grandes beneficios en performance, modularidad y promueve un flujo de datos y eventos, facilitando la planeación y desarrollo de apps complejas (Dorantez, 2015).

Javascript cuenta también con entornos de ejecución:

- Node.js: es un entorno Javascript del lado del servidor, basado en eventos. Node ejecuta JavaScript utilizando el motor V8, desarrollado por Google para uso de su navegador Chrome. Aprovechando el motor V8 permite a Node proporciona un entorno de ejecución del lado del servidor que compila y ejecuta JavaScript a velocidades increíbles. El aumento de velocidad es importante debido a que V8 compila Javascript en código de máquina nativo, en lugar de interpretarlo o ejecutarlo como bytecode. Node es de código abierto, y se ejecuta en Mac OS X, Windows y Linux. (NetConsulting , 2015).

#### **4.3.1.1 Ant Design**

Ant Design es una librería de React que cuenta con muchos componentes fáciles de usar, útiles para construir elegantes interfaces de usuario. (Taylor, 2018)

Creado por el conglomerado chino, Alibaba, Ant Design es utilizado en grandes empresas: Alibaba, Tencent, Baidu, entre otras. (Taylor, 2018) Su objetivo es uniformar las especificaciones de la interfaz de usuario para los proyectos internos, reducir el costo innecesario de las diferencias de diseño e implementación y liberar los recursos de diseño y desarrollo front-end. (Ant Design, 2019)

Aprovechando las especificaciones unitarias, Ant Design hace que el diseño y el prototipo sean más simples y accesibles para todos los miembros del

proyecto, lo que promueve de manera integral la experiencia y la eficiencia de desarrollo de aplicaciones y productos de fondo. (Ant Design, 2019)

#### 4.3.2 Tecnologías Backend

(Culturación, s.f.) explica que el back-end

*“... es la parte que procesa la entrada de datos que se efectuó desde el front-end es decir, son los procesos que utiliza el administrador del sitio con sus respectivos sistemas para resolver las peticiones de los usuarios. De esta manera en manera conjunta el front-end y el back-end interactúan en un sistema web o software para resolver las necesidades de los usuarios.”*

##### 4.3.2.1 ORM: Object Relational Mapping

El mapeo objeto-relacional es una técnica de programación para convertir datos del sistema de tipos utilizado en un lenguaje de programación orientado a objetos al utilizado en una base de datos relacional. En la práctica esto crea una base de datos virtual orientada a objetos sobre la base de datos relacional. Esto posibilita el uso de las características propias de la orientación a objetos (esencialmente la herencia y el polimorfismo).

Las bases de datos relacionales solo permiten guardar tipos de datos primitivos (enteros, cadenas de texto, etc.) por lo que no se pueden guardar de forma directa los objetos de la aplicación en las tablas, sino que estos se deben de convertir antes en registros, que por lo general afectan a varias tablas. En el momento de volver a recuperar los datos, hay que hacer el proceso contrario, se deben convertir los registros en objetos. Es entonces cuando ORM cobra importancia, ya que se encarga de forma automática de convertir los objetos en registros y viceversa, simulando así tener una base de datos orientada a objetos.

Entre las ventajas que ofrecen los ORM se encuentran: rapidez en el desarrollo, abstracción de la base de datos, reutilización, seguridad, mantenimiento del código, lenguaje propio para realizar las consultas. (Yanes Enriquez & Gracia del Busto, 2011)

#### **4.3.2.2 Base de datos no relacionales**

De acuerdo con (Amazon Web Services, Inc., 2019) las bases de datos No SQL están diseñadas específicamente para modelos de datos específicos y tienen esquemas flexibles para crear aplicaciones modernas. Las bases de datos No SQL son ampliamente reconocidas porque son fáciles de desarrollar, su funcionalidad y el rendimiento a escala. Usan una variedad de modelos de datos, que incluyen documentos, gráficos, clave-valor, en-memoria y búsqueda.

Utilizan una variedad de modelos de datos para acceder y administrar datos, como documentos, gráficos, clave-valor, en-memoria y búsqueda. Estos tipos de bases de datos están optimizados específicamente para aplicaciones que requieren grandes volúmenes de datos, baja latencia y modelos de datos flexibles, lo que se logra mediante la flexibilización de algunas de las restricciones de coherencia de datos en otras bases de datos.

Las bases de datos No SQL se adaptan perfectamente a muchas aplicaciones modernas, como dispositivos móviles, web y juegos, que requieren bases de datos flexibles, escalables, de alto rendimiento y altamente funcionales para proporcionar excelentes experiencias de usuario.

- Flexibilidad: las bases de datos No SQL generalmente ofrecen esquemas flexibles que permiten un desarrollo más rápido y más iterativo. El modelo de datos flexible hace que las bases de datos No SQL sean ideales para datos semiestructurados y no estructurados.
- Escalabilidad: las bases de datos No SQL generalmente están diseñadas para escalar usando clústeres distribuidos de hardware en lugar de escalar añadiendo servidores caros y sólidos. Algunos proveedores de la nube manejan estas operaciones fuera del alcance, como un servicio completamente administrado.
- Alto rendimiento: la base de datos No SQL está optimizada para modelos de datos específicos (como documentos, clave-valor y gráficos) y patrones de acceso que permiten un mayor rendimiento que el intento de lograr una funcionalidad similar con bases de datos relacionales.



- Altamente funcional: las bases de datos No SQL proporcionan API altamente funcionales y tipos de datos que están diseñados específicamente para cada uno de sus respectivos modelos de datos.

#### 4.3.2.3 MongoDB

MongoDB es una base de datos open-source desarrollada por 10gen en C++. Fue lanzada en el año 2009 y pertenece a la categoría de las bases de datos documentales, esta base de datos surge como una nueva tendencia en el desarrollo de bases de datos y se refieren en general, a las bases de datos sin un esquema fijo. Además, suelen tener una seguridad de transacciones a un nivel más bajo, pero son más rápidos en el acceso a los datos y escalan mejor que las bases de datos relacionales. (Suter, 2012)

La base de datos MongoDB consiste en un conjunto de bases de datos en la que cada base de datos contiene varias colecciones y cada colección puede contener diferentes tipos de objetos y cada objeto también llamado documento se representa como una estructura JSON que es una lista de pares de clave-valor (esto se debe a que MongoDB trabaja con esquemas dinámicos. El valor puede ser de tres tipos: un valor primitivo, un array de documentos, o de nuevo una lista de pares de clave-valor o documentos como se muestra en la siguiente figura: (Suter, 2012)

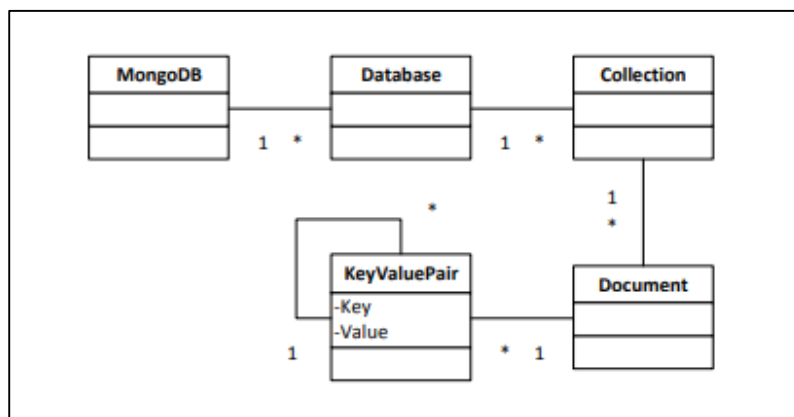


Imagen 2 Tipos de colecciones en una base de datos no relacional

MongoDB posee una terminología diferente a la de una base de datos relacional:

- Colecciones: Las "colecciones" en Mongo son equivalentes a las tablas en bases de datos relacionales. Pueden contener varios documentos JSON.
- Documentos: Los "documentos" son equivalentes a registros o filas de datos en SQL. Mientras que una fila de SQL puede hacer referencia a datos en otras tablas, los documentos de Mongo generalmente combinan eso en un documento.
- Campo: Los "campos" o los atributos son similares a las columnas en una tabla SQL.
- Esquema: Si bien Mongo no tiene esquema, SQL define un esquema a través de la definición de la tabla. Un "esquema" de Mongoose es una estructura de datos de documento (o forma del documento) que se aplica a través de la capa de aplicación.
- Modelos: Los "modelos" son constructores de orden superior que toman un esquema y crean una instancia de un documento equivalente a los registros en una base de datos relacional. (Karnik, 2018)

#### **4.3.2.4 Mongoose**

Mongoose es una biblioteca de modelado de datos de objetos (ODM en inglés "Object Data Modeling") para MongoDB y Node.js. Administra las relaciones entre los datos, proporciona la validación del esquema y se utiliza para traducir entre los objetos en el código y la representación de esos objetos en MongoDB.

Un modelo de Mongoose es una envoltura en el *schema* de Mongoose. Un *schema* de Mongoose define la estructura del documento, los valores predeterminados, los validadores, etc., mientras que un modelo de Mongoose proporciona una interfaz a la base de datos para crear, consultar, actualizar, eliminar registros, etc. (Karnik, 2018)

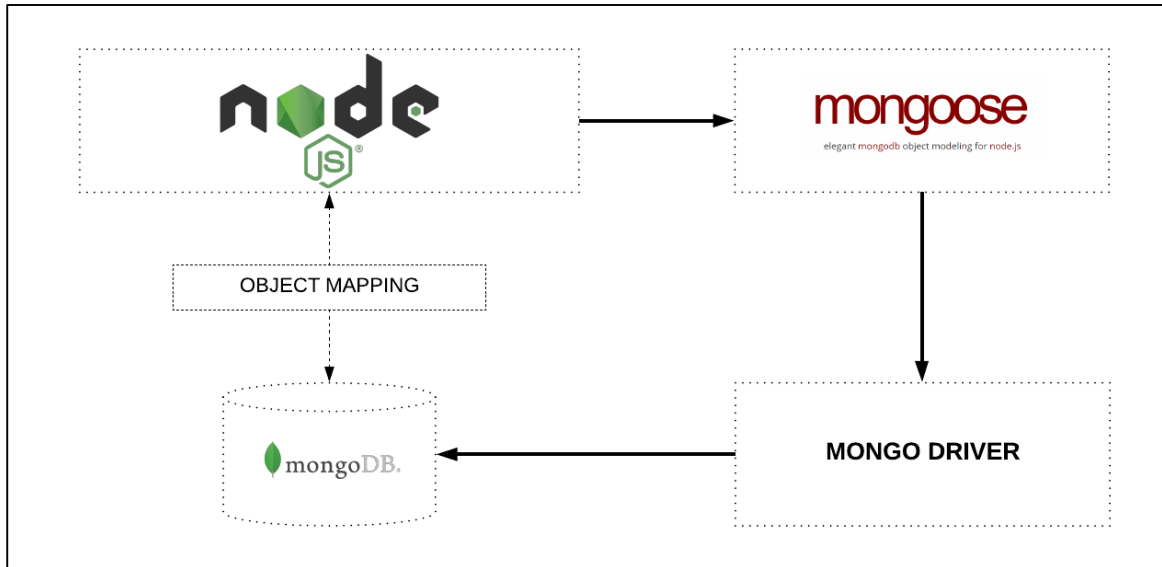


Imagen 3 Mapeo de objetos entre Node y MongoDB

Un *schema* para una base de datos es similar a una "clase" en la programación orientada a objetos. Cuando una clase es como un plano para crear objetos en un programa, un *schema* es un plano para crear objetos (MongoDB los llama "documentos") en una base de datos. Le decimos a la base de datos cómo se estructurarán los documentos que vamos a insertar (tendrá una propiedad de nombre que será de tipo "String", una propiedad de administrador que será de tipo "booleano", etc.), y la base de datos puede hacer alguna validación por nosotros (la base de datos validará si cumple con el requisito del *schema* y si debe agregarlo o no). (Ziroll, 2015)

#### 4.3.2.5 mLab

mLab es una base de datos como servicio (o DBaaS por sus siglas en inglés "DataBase-as-a-Service") que hospeda MongoDB. Las DBaaS, consisten en ofrecer bases de datos como servicio en demanda, con o sin imagen de máquina virtual y facilitando cargas que residen de manera segura. Es un servicio de pago en el que la base de datos se ejecuta en la infraestructura física de nuestro proveedor de servicios. (mLab, 2019)

Según (Grupo Garutu. IT Solutions, s.f.), las DBaaS ofrecen los siguientes beneficios:

- Eliminación de la infraestructura física. En un entorno de base de datos en la nube, el proveedor de computación en nube de servidores, almacenamiento y otras infraestructuras es responsable del mantenimiento y la disponibilidad. En un entorno DBaaS, el proveedor de servicios es responsable de mantener y operar el software de base de datos, dejando a los usuarios DBaaS responsables sólo de sus propios datos.
- Ahorro de costes. A través de la eliminación de una infraestructura física y operada por un departamento de TI, pueden lograrse ahorros significativos a partir de gastos de capital reducidos, menos personal, disminución del coste en electricidad y una menor cantidad de espacio físico requerido.
- Escalabilidad instantánea. Si el tamaño de la base de datos tuviera una demanda variable debido a picos de negocios estacionales o picos inesperados en la demanda, nuestro proveedor podría ofrecernos rápidamente la capacidad adicional basada en tarifas, el rendimiento y nuestro ancho de banda. En una infraestructura tradicional, probablemente necesitaría esperar semanas o meses para la adquisición e instalación de recursos adicionales de servidor, almacenamiento o comunicaciones.
- Experiencia especializada. Hoy en día encontrar a expertos en bases de datos puede ser difícil, llegando a resultar prohibitivo mantenerlos como parte de nuestro personal. En un entorno DBaaS, nuestro proveedor puede atender a miles de clientes y por tanto, mantener y retener ese talento es posible económicamente para ellos, teniendo nosotros acceso a la experiencia de gente experta en la materia a nivel mundial.
- Precios decrecientes. Con los grandes y continuos avances en tecnología, dentro de un mercado intensamente competitivo entre los principales proveedores de servicios, los precios de la llamada computación en la nube se someten a una re-calibración continua.

Por tanto, esta disminución en los precios es un importante impulso para migrar nuestras tradicionales bases de datos a la nube, y obtener así, sus enormes ventajas. (Grupo Garutu. IT Solutions, s.f.)

#### **4.3.2.6 Nginx**

Nginx es un software de código abierto que trabaja como servidor web, puede ser configurado como servidor proxy inverso, es decir, se configura como front-end de otro servidor más lento, de manera que sirve en primer lugar el contenido estático reduciendo el tiempo de carga del servidor en back-end, puede también trabajar para balancear la carga de los servidores back-end o para almacenar en cache el contenido de un servidor más lentos.

#### **4.3.2.7 GraphQL**

GraphQL es un protocolo de consulta de datos originalmente elaborado por Facebook para uso interno. Es un lenguaje de consultas que te permite definir qué datos pedirle a un API. El objetivo esencial de GraphQL es ofrecer a los clientes una forma más directa, sencilla y eficiente para obtener exactamente los datos que requieren, a través de un protocolo potente y dinámico. (Apilyca, 2018).

GraphQL no es una librería ni un framework, se trata de un lenguaje de consulta que puede ser utilizado indistintamente en diferentes códigos de programación, incluidos algunos de los más usados hoy en día como JavaScript, Ruby, Java, PHP y Python.

Una de las principales cualidades de GraphQL es que se cuenta con un esquema muy sólido, confiable y bien definido que permite tener el control de las configuraciones. Hay 3 operaciones principales: las consultas a la API, las mutaciones (que son funciones que realizan modificaciones y mandan una respuesta) y las suscripciones (que permiten recibir los datos de los cambios en el servidor). Todo esto se realiza bajo los parámetros bien definidos del SDL.

GraphQL cuenta con las siguientes características:

- En ocasiones solo se necesita un único campo de un recurso, pero el servidor devuelve todos, sin embargo, con GraphQL

- Es la aplicación la que tiene el control del recurso, reduciendo el ancho de banda y los tiempos.
- Con GraphQL se puede acceder a distintos recursos en una única llamada.
- Se elimina el concepto de *end-points*, sustituyéndolo por tipos con los indicar solo lo que se necesita.
- Se intenta eliminar el versionado de las APIs pudiendo deprecarse el uso de atributos.
- GraphQL desacopla la dependencia con el acceso a las bases de datos.

Como servidor GraphQL se utiliza Express, que es un marco de aplicación web Node.js mínimo y flexible que proporciona un conjunto sólido de funciones para aplicaciones web y móviles. (Express, 2019).

#### **4.3.2.8      *Apollo Client***

Apollo Client es un cliente GraphQL de almacenamiento en caché con todas las funciones con integraciones para React, Angular y más. Le permite crear fácilmente componentes de la interfaz de usuario que obtienen datos a través de GraphQL. Para obtener el máximo valor de Apollo Client, debe usarlo con una de sus integraciones de capa de vista. Apollo Client también tiene integraciones de capas de vista para todos los marcos front-end populares. Se puede utilizar en cualquier front-end de JavaScript en el que desee utilizar datos de un servidor GraphQL. Es:

- Adoptable en forma incremental, para que pueda colocarlo en una aplicación JavaScript existente y comenzar a usar GraphQL para solo una parte de su interfaz de usuario.
- Compatible universalmente, por lo que Apollo funciona con cualquier configuración de compilación, cualquier servidor GraphQL y cualquier esquema GraphQL.
- Es fácil comenzar con él, por lo que puede comenzar a cargar datos de inmediato y aprender sobre funciones avanzadas más adelante.

- Inspeccionable y comprensible, para que pueda tener excelentes herramientas de desarrollador para comprender exactamente lo que está sucediendo en su aplicación.
- Creado para aplicaciones interactivas, para que sus usuarios puedan realizar cambios y verlos reflejados en la interfaz de usuario de inmediato.  
Pequeño y flexible, para que no tengas cosas que no necesitas. El núcleo está bajo 25kb comprimido.
- Impulsado por la comunidad, porque Apollo está impulsado por la comunidad y sirve una variedad de casos de uso. Todo está planeado y desarrollado al aire libre. (GitHub, 2019)

#### **4.3.2.9      Servidor de Correo: SendGrid**

Según (Microsoft Azure Contributors, 2014), SendGrid es un servicio de correo electrónico basado en la nube que proporciona entrega confiable de correo electrónico transaccional, escalabilidad y análisis en tiempo real junto con API flexibles que facilitan la integración personalizada. Los escenarios comunes de uso de SendGrid incluyen:

- Envío automático de recibos a clientes.
- Administración de listas de distribución para enviar a los clientes volantes electrónicos mensuales y ofertas especiales.
- Recopilación de métricas en tiempo real para cosas como correo electrónico bloqueado y capacidad de respuesta del cliente.
- Generación de informes para ayudar a identificar tendencias.
- Reenviar consultas de clientes.
- Correo electrónico y notificaciones desde su aplicación

#### **4.3.2.10 Integración Continua / Implementación continua<sup>2</sup>**

CI/CD es un método para entregar con frecuencia aplicaciones a los clientes mediante la introducción de la automatización en las etapas de desarrollo de la aplicación. Los principales conceptos atribuidos a CI/CD son la integración continua, la entrega continua y la implementación continua. CI/CD es una solución a los problemas que puede causar la integración de un nuevo código para los equipos de desarrollo y operaciones. Específicamente, CI/CD introduce la automatización continua y el monitoreo continuo a lo largo del ciclo de vida de las aplicaciones, desde las fases de integración y prueba hasta la entrega y el despliegue. Tomadas en conjunto, estas prácticas conectadas a menudo se denominan "canalización de CI/CD" y están respaldadas por equipos de desarrollo y operaciones que trabajan juntos de manera ágil.

#### **4.3.2.11 Amazon Web Services (AWS)**

De acuerdo con (TIC Portal, 2019), Amazon Web Services, también conocida como AWS, es un conjunto de herramientas y servicios de *Cloud Computing* de Amazon. Este servicio se lanzó oficialmente en 2006 y para junio de 2007 AWS ya contaba con una base de usuarios de aproximadamente 180 mil personas. Entre las empresas que la utilizan se encuentran algunas como Reddit, Foursquare, Pinterest, Netflix, la NASA o la CIA, y algunas españolas como Mapfre, el FC Barcelona o Interflora. Esto se debe principalmente a la madurez del servicio frente a otros similares y las posibilidades que ofrece el amplio abanico de herramientas disponibles.

Amazon Web Services ofrece herramientas en las siguientes categorías:

- *Cloud Computing*: todo lo necesario para la creación de instancias y el mantenimiento o el escalado de las mismas. Amazon EC2 es el rey indiscutible dentro de los servicios de computación en la nube de Amazon.
- Bases de datos: distintos tipos de bases de datos pueden permanecer en la nube mediante el servicio Amazon RDS, que incluye distintos tipos a

---

<sup>2</sup> Referido en adelante como CI/CD (Continuous integration / Continuous deployment) por sus siglas en inglés.



elegir como MySQL, PostgreSQL, Oracle, SQL Server y Amazon Aurora, o Amazon DynamoDB para No SQL.

- Creación de redes virtuales: permite la creación de redes privadas virtuales a través de la nube, gracias principalmente al servicio Amazon VPC.
- Aplicaciones empresariales: Amazon WorkMail es el servicio de correo empresarial que ofrece Amazon, al que pueden unirse otros servicios como Amazon WorkDocs y Amazon WorkSpaces.
- Almacenamiento y gestores de contenido: tipos de almacenamiento diferentes, tanto para archivos con acceso regular, poco frecuente o incluso como archivo. Amazon S3 es el servicio principal, aunque complementan la oferta otros como Amazon Glacier o Amazon EBS.
- Inteligencia de negocios o Business Intelligence (BI): sistemas para análisis de datos empresariales a gran escala y otros servicios para la gestión de flujos de datos.
- Gestión de aplicaciones móviles: herramientas como Amazon Mobile Hub permiten la gestión, creación, testeo y mantenimiento de aplicaciones móviles a través de la nube.
- Internet de las cosas (IoT, Internet of Things): para establecer conexiones y análisis de todos los dispositivos conectados a internet y los datos recogidos por los mismos.
- Herramientas para desarrolladores: para almacenar código, implementarlo automáticamente o incluso publicar software mediante un sistema de entrega continua.
- Seguridad y control de acceso: se pueden establecer autenticaciones en varios pasos para poder proteger el acceso a sus sistemas internos, ya estén en la nube o instalados de forma local en sus instalaciones.

#### **4.3.2.12 Heroku**

Heroku es una plataforma como servicio que permite a las empresas crear, entregar, monitorear y escalar aplicaciones. Permite manejar los servidores y sus configuraciones, escalamiento y la administración. Se pueden alojar en él, aplicaciones de diferentes lenguajes de programación como Python, Java, PHP y

más. Los servicios que Heroku ofrece están claramente orientados para desarrolladores, nuevas y pequeñas empresas. Heroku nació en el 2007 y fue adquirida en el 2010 por salesforce.com. (Platzi, 2017)

Heroku utiliza contenedores Linux (Ubuntu) los cuales son llamados “dynos”, estos son utilizados para alojar las aplicaciones web, servicios web o aplicaciones que se ejecutan del lado del servidor, así mismo cuenta con la posibilidad de instalar add-ons para agregar funcionalidades a dichos contenedores. (Urrutia, 2018)

#### 4.4. Seguridad

(Mier y Terán & Vasquez Martínez, 2016) sostienen que se deben tomar en cuenta los siguientes puntos en cuanto a seguridad se refiere:

- Rastrear el paso de los datos: conocer los pasos que ha recorrido la información en todo momento. De dónde vienen los datos y hacia dónde van.
- Filtrar entradas: es el proceso por el cual se prueba la validez de los datos. Si nos aseguramos que los datos son filtrados apropiadamente al entrar, se reduce el riesgo de que datos contaminados sean usados para provocar funcionamientos no deseados a la aplicación (Lázaro, Filtrado de datos de entrada en PHP, 2016).
- Escapado de salidas (*output escaping*): codificar o decodificar caracteres especiales de tal forma que su significado sea preservado. Se tiene que identificar las salidas, escapar las salidas y distinguir entre datos escapados y no escapados (Lázaro, Escape de datos de salida en PHP, 2016).

Existen formas de prevenir ataques en la seguridad de la aplicación, a continuación, se plantean posibles ataques y sus posibles soluciones:

**Ataques XSS** (*Cross-Site Scripting* por sus siglas en inglés), estos ocurren cuando un atacante es capaz de inyectar un script, normalmente JavaScript, en el output de una aplicación web de forma que se ejecuta en el navegador del cliente.

Para prevenir estos ataques se recomienda tomar tres medidas principales (Lázaro, Ataques XSS: Cross-Site Scripting en PHP, 2016):

- Validación de datos: es el proceso de asegurar que la aplicación analiza el tipo de datos correcto.
- Sanitización de datos: se centra en manipular los datos para asegurarse que son seguros, eliminando cualquier parte indeseable y normalizándolos en la forma correcta.
- Escapado de salidas: Para proteger la integridad de los datos que se devuelven, el output data, se debe escapar cualquier dato que se devuelve al usuario. Esto evita que el navegador malinterprete alguna secuencia especial de caracteres.

El **Intercambio de Recursos de Origen Cruzado (CORS)** por sus siglas en inglés) es un mecanismo que utiliza cabeceras HTTP adicionales para permitir que un usuario obtenga permiso para acceder a recursos seleccionados desde un servidor, en un origen distinto (dominio) al que pertenece. Un agente crea una petición HTTP de origen cruzado cuando solicita un recurso desde un dominio distinto, un protocolo o un puerto diferente al del documento que lo generó.

Un ejemplo de solicitud de origen cruzado: el código JavaScript front-end de una aplicación web que se localiza en `http://domain-a.com` utiliza XMLHttpRequest para cargar el recurso `http://api.domain-b.com/data.json`.

Por razones de seguridad, los exploradores restringen las solicitudes HTTP de origen cruzado iniciadas dentro de un script. Por ejemplo, XMLHttpRequest y la API Fetch siguen la política de mismo-origen. Esto significa que una aplicación que utilice esas APIs XMLHttpRequest sólo puede hacer solicitudes HTTP a su propio dominio, a menos que se utilicen cabeceras CORS.

El W3C Grupo de Trabajo de Aplicaciones Web recomienda el nuevo mecanismo de Intercambio de Recursos de Origen Cruzado (CORS, por sus siglas en inglés). CORS da controles de acceso a dominios cruzados para servidores web y transferencia segura de datos en dominios cruzados entre navegadores y

servidores Web. Los exploradores modernos utilizan CORS en un contenedor API (como XMLHttpRequest o Fetch) para ayudar a mitigar los riesgos de solicitudes HTTP de origen cruzado. (Comunidad Developer Mozilla, 2019)

Una forma de prevenir ataques y proteger el sitio web es utilizando el protocolo **Capa de conexión segura SSL** (*Secure Sockets Layer* por sus siglas en inglés), que es un protocolo diseñado para transmitir información de ida y de manera segura hacia atrás. Las aplicaciones que utilizan el protocolo SSL saben cómo dar y recibir claves de cifrado con otras aplicaciones, así como la manera de cifrar y descifrar los datos enviados entre los dos (Digicert, 2003).

El protocolo SSL fue diseñado con el objeto de proveer privacidad y confiabilidad a la comunicación entre dos aplicaciones. Las conexiones realizadas por medio de este protocolo tienen las siguientes propiedades básicas (Roel, s.f.):

- **Privada:** Después de un proceso inicial de "*handshake*" en el cual se define una clave secreta, se envía la información encriptada por medio de algún método simétrico (DES, RC4).
- **Segura:** La identidad de cada extremo es autenticada usando métodos de cifrado asimétricos o de clave pública (RSA, DSS).
- **Confiable.** El transporte del mensaje incluye un control de la integridad del mismo usando una MAC cifrada con SHA y MD5.

## 4.5. Estándares web

Los estándares Web son reglas agrupadas que propician ser sistemas más usables y accesibles. Se basan en tres aspectos fundamentales:

1. La interacción con el usuario, buscando una comunicación armoniosa entre el mismo y la aplicación.
2. La optimización del modo en que se le presentan los datos al usuario, ayudando a que conozca en qué parte del ambiente de trabajo está situado y en qué momento.
3. La estructura del sistema optimiza la ubicación de los componentes y las distintas secciones que se definen.

Así se logra una óptima navegabilidad debido a su aplicación, aprovechando al máximo el área de contenido; de modo que la sección que requiera mayor interacción con el usuario sea la priorizada en cuanto a espacio, agrupamiento y elegancia.

Los estándares Web son definidos por el grupo W3 (*World Wide Consortium*), una comunidad internacional creada en 1994 por Tim Berners-Lee. Son las respuestas más eficaces a la rápida y continua evolución tecnológica que experimenta la red. Adecuarse a ellos hace posible que el trabajo de hoy constituya una base efectiva en el futuro y ayude a evolucionar tecnológicamente con el medio.

Según su página web oficial (W3, 2017), los estándares están definidos en:

- **Diseño web y aplicaciones:**

El diseño web y las aplicaciones incluyen los estándares para crear y renderizar páginas web, incluidos HTML, CSS, SVG, API de dispositivos y otras tecnologías para aplicaciones web ("aplicaciones web"). También incluye información sobre cómo hacer que las páginas sean accesibles para personas con discapacidades (WCAG), para internacionalizarlas y hacer que funcionen en dispositivos móviles.

- **Web de dispositivos:**

W3C se enfoca en tecnologías para permitir el acceso a la Web en cualquier lugar y en cualquier momento, utilizando cualquier dispositivo. Esto incluye el acceso a la Web desde teléfonos móviles y otros dispositivos móviles, así como el uso de la tecnología Web en electrónica de consumo, impresoras, televisión interactiva e incluso automóviles.

- **Arquitectura web:**

La arquitectura web se centra en las tecnologías y los principios fundamentales que sustentan la web, incluidos los URI y HTTP.

- **Web semántica:**

Además de la clásica "Web de documentos", W3C está ayudando a construir una pila de tecnología para admitir una "Web de datos", el tipo de datos que se encuentran en las bases de datos. El objetivo final de la Web de datos es permitir a las computadoras realizar un trabajo más útil y desarrollar sistemas que puedan admitir interacciones de confianza a través de la red.

El término "Web semántica" se refiere a la visión de W3C de la Web de los datos vinculados. Las tecnologías de la Web semántica permiten a las personas crear almacenes de datos en la Web, crear vocabularios y escribir reglas para manejar los datos. Los datos vinculados están habilitados por tecnologías como RDF, SPARQL, OWL y SKOS.

- **Tecnología XML:**

Tecnologías XML que incluyen XML, espacios de nombres XML, esquema XML, XSLT, intercambio eficiente de XML (EXI) y otros estándares relacionados.

- **Web de servicios:**

Web de servicios se refiere al diseño basado en mensajes que se encuentra con frecuencia en la web y en el software empresarial. La Web de servicios se basa en tecnologías como HTTP, XML, SOAP, WSDL, SPARQL y otras.

- **Navegadores y herramientas de autoría:**

La utilidad y crecimiento de la web depende de su universalidad. Deberíamos poder publicar independientemente del software que usemos, la computadora que tengamos, el idioma que hablemos, ya sea que estemos cableados o inalámbricos, independientemente de nuestros modos sensorial o de interacción. Deberíamos poder acceder a la web desde cualquier tipo de hardware que pueda conectarse a Internet, ya sea fijo o móvil, pequeño o grande. W3C facilita esta escucha y mezcla a través de estándares web internacionales. Estos estándares aseguran que toda la locura brillante continúe mejorando una web que está abierta para todos nosotros.

## **4.6. Redes**

### **4.6.1 RADIUS**

Según (García, 2012) hace referencia al protocolo y servidor RADIUS:

RADIUS (*Remote Authentication Dial-In User Server* por sus siglas en inglés) es un protocolo cliente/servidor, donde el cliente es un NAS (Network Access Server) y el servidor es un software ejecutado en un equipo UNIX, LINUX o Windows. Como protocolo de transporte emplea UDP, para establecer comunicación utiliza dos puertos: el 1813 para contabilidad y el 1812 para autenticación y autorización.

### **4.6.2 Servidor RADIUS**

Es un software instalado como servicio en el sistema operativo de una computadora, es el encargado de administrar las cuentas de acceso. Se utiliza principalmente por los proveedores de servicios de Internet para gestionar acceso a Internet a sus clientes. Recibe la autenticación y luego de realizar la comparación

con sus registros envía un mensaje permitiendo o negando el acceso, además ira almacenando los eventos de dichos procesos. Para aceptar las consultas del cliente debe tener un perfil del NAS con la dirección IP del cliente y la clave de autorización.

#### 4.6.3 Portal Cautivo

En primer lugar, hay que aclarar que este sistema de control de acceso necesita un servidor que realice el control de la conexión al exterior (ver Ilustración 2) (Castro, 2005). Con este método de autenticación, el cliente consigue conectarse a la WIFI sin problemas, e incluso se le asigna una IP, pero sin capacidad de comunicarse fuera del entorno que se haya previamente definido.

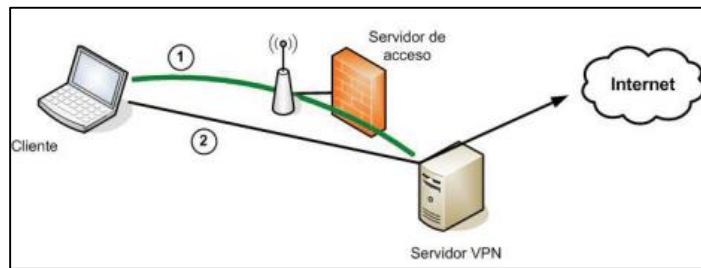


Imagen 4 Fuente: (Castro, 2005)

Cuando el cliente intenta establecer comunicación con una página web externa, automáticamente es redirigido a un portal cautivo en el cual puede autenticarse. Una vez hecho, ya sea con un simple usuario/clave o previo pago de una tarjeta de conexión o incluso con una tarjeta de crédito, el usuario consigue conectividad a Internet durante el tiempo establecido (Castro, 2005). Para poder mantener la sesión viva, el cliente tiene que mantener una ventana del navegador “pop-up” abierta que se encarga de comunicar automáticamente y cada cierto tiempo con el portal de acceso, garantizando que la conexión siga abierta.

Existen numerosos ejemplos de este tipo de sistema que ahora mismo está operando en todo el mundo. Su utilización más habitual corresponde con accesos WIFI en espacios públicos “hotspots”. Esto se debe a que el navegador es una herramienta muy extendida, la mayoría de dispositivos con acceso WIFI tienen un navegador, y su utilización resulta muy natural para los usuarios, permitiendo así mismo el pago del servicio (conexión a Internet) de una manera fácil.



## 4.7. *Extreme Programming (XP)* como metodología de desarrollo

Extreme Programming<sup>3</sup> (Programación Extrema, XP por su siglas en inglés) es una metodología ágil de desarrollo de software. Las primeras ideas y métodos asociadas con XP se produjeron a finales de 1980. Sin embargo, la base principal fue escrita por Kent Beck.

Según (Pressman, 2010),

*“Beck define un conjunto de cinco valores que establecen una base para todo trabajo realizado como parte de XP: comunicación, simplicidad, retroalimentación, valor y respeto” (p. 72).*

Para lograr la **comunicación**, XP enfatiza la comunicación estrecha e informal (verbal) colaboración entre los clientes y los desarrolladores, el establecimiento de metáforas<sup>4</sup> para comunicar conceptos importantes, retroalimentación continua, y evitar la voluminosa documentación como medio de comunicación.

La **simplicidad** se logra restringiendo a los desarrolladores a diseñar únicamente las necesidades inmediatas en lugar de las necesidades futuras. El desarrollador puede después mejorar la estructura interna de un diseño, sin alterar su funcionalidad externa o comportamiento.

Siguiendo con los valores de XP, continúa definiendo la **retroalimentación**, diciendo que

*“... ésta se deriva de tres fuentes: el software implementado en sí, el cliente y otros miembros del equipo de desarrollo. Diseñando e implementando estrategias de pruebas, el software provee al equipo de retroalimentación. XP hace uso de la unidad de prueba como principal táctica. Cada que se desarrolla una clase, el equipo de desarrollo crea una unidad de prueba para practicar su operación de acuerdo a su funcionalidad. Como parte*

---

<sup>3</sup> Referida en adelante como: metodología XP o XP.

<sup>4</sup> Metáfora en XP se refiere a una historia que cualquiera (desarrolladores, clientes y gerentes) pueden decir acerca de cómo funciona el sistema (Pressman, 2010).

de la retroalimentación, se entregan al usuario las historias de usuario o casos de uso, las que son utilizadas como base para las pruebas de aceptación.” (p. 73).

Beck, propone otra base de la metodología XP y es el **valor**. Sin embargo, (Pressman, 2010) propone otro nombre para ello y es la **disciplina**. Explica que los desarrolladores se frustran al intentar diseñar para el futuro, pero que un equipo de desarrollo que utiliza XP debe tener la disciplina de diseñar para el hoy, reconociendo que los requerimientos pueden cambiar drásticamente en el futuro.

La metodología XP utiliza una serie de reglas y prácticas que ocurren en el contexto de cuatro marcos de referencia: planificación, diseño, codificación y pruebas. El proceso se muestra en la siguiente figura:

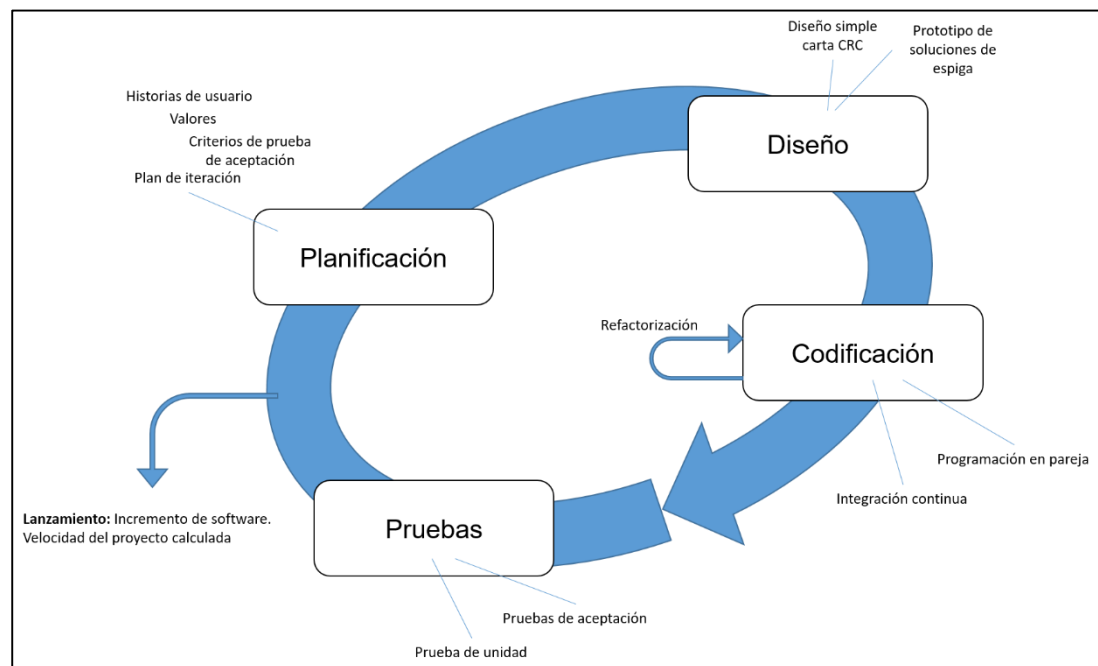


Imagen 5 Fases de la metodología X (Pressman, 2010) Traducción propia

La fase de **planificación** empieza recopilando las historias de usuario. Los programadores las evalúan el tiempo de desarrollo de cada una, si alguna presenta riesgos, se realizan pequeños programas de prueba (“spikes”). Se acuerda un cronograma con el equipo de trabajo (Joskowicz, 2008).

**Historia de usuarios:** éstas sustituyen a los “casos de uso”. Son escritas por el cliente, en su propio lenguaje. Deben tener el detalle mínimo como para que

los programadores puedan realizar una estimación poco riesgosa del tiempo que llevará su desarrollo. Cuando llegue el momento de la implementación, los desarrolladores dialogarán con el cliente para obtener detalles necesarios (Joskowicz, 2008).

(Pressman, 2010) sostiene que XP hace énfasis en que la fase de **diseño** debe mantener los diseños simples y claros.

- **Simplicidad:** XP propone implementar el diseño más simple posible que funcione. Se sugiere nunca adelantar la implementación de funcionalidades que no correspondan a la iteración en la que se está trabajando.
- **Soluciones “*spike*”:** son pequeños programas de prueba llamados “*spike*” para explorar diferentes funciones. Son únicamente usados para probar o evaluar una solución.
- **Recodificación:** consiste en escribir nuevamente una parte del código del programa sin cambiar la funcionalidad, a los efectos de hacerlo más simple, conciso y entendible.
- **Metáforas:** una frase que todos entiendan sin necesidad de explicaciones, para saber de qué función o parte del programa se está hablando sin necesidad de tecnicismos.
- **Tarjetas CRC:** (*Class, Responsibilities and Collaboration*) sirven para diseñar el sistema en conjunto entre todo el equipo. Permiten reducir el modo de pensar procedural y apreciar la tecnología de objetos

En la fase **codificación** se necesita la disponibilidad del cliente en todo el proyecto, formando parte del equipo. Se hace uso de la programación en pareja, donde dos programadores están frente a la computadora codificando. Luego se integra con los demás desarrolladores.

Por último, la fase de **pruebas** donde se crean pruebas automáticas con un framework. Se valida cada prueba y su solución. Por último, XP propone las pruebas de aceptación donde los clientes son los que realizan las pruebas y

comprueban la funcionalidad del sistema, éstas son derivadas de las pruebas de usuario.

XP se basa en doce prácticas básicas que deben seguirse para que funcione de manera eficiente. La (Universidad Unión Bolivariana, 2008) resume dichas prácticas:

- Todas las personas involucradas forman parte del equipo.
- Planificar las historias de usuario en orden de realización.
- Proponer pruebas para validar las versiones.
- Crear versiones pequeñas del software que ofrezcan algo útil para verlo funcionar.
- Mantener el código sencillo y crear un diseño simple.
- Programar en pareja.
- Realizar programas de prueba automática y ejecutarlas con frecuencia.
- Tener un ejecutable del proyecto que funcione y en cuanto se tenga una nueva función, recompilarse y probarse.
- El código es de todos. Cualquiera puede tocar y conocer cualquier parte del código.
- Trabajar a un ritmo sostenible
- Seguir los mismos estándares de programación.
- Crear metáforas, es decir, frases o nombres que definan cómo funcionan las distintas partes del programa.



# CAPITULO III:

## Análisis y Presentación de Resultados

- Estudio de factibilidad
- Aplicación de la metodología XP

## V. ESTUDIO DE FACTIBILIDAD

### 5.1. Aspectos Técnicos

#### 5.1.1 Inventario de recursos de hardware y software

En esta sección se describe el inventario de recursos tecnológicos que posee Wimix y se plantean otras herramientas tecnológicas necesarias para el desarrollo e implementación del sistema.

En la Tabla 1 Inventario de recursos tecnológicos de Wimix Solutions and Services) se muestran los recursos tecnológicos que Wimix posee al momento:

| Recurso  | Cantidad                   | Características  |
|--|----------------------------|--|
| <b>Almacenamiento de objetos y datos</b>       | Pago mensual               | <ul style="list-style-type: none"> <li>Amazon Web Service S3</li> </ul>  |
| <b>Sistema de gestión de puntos de accesos</b> | Pago mensual               | <ul style="list-style-type: none"> <li>HotspotSystem</li> </ul>  |
| <b>Dominio</b>                                 | Pago anual                 | <ul style="list-style-type: none"> <li>JustHost</li> </ul>   |
| <b>Protocolo de seguridad</b>                  | Pago anual                 | <ul style="list-style-type: none"> <li>Certificado SSL</li> </ul>  |
| <b>Certificado White Label</b>                 | Pago único                 | <ul style="list-style-type: none"> <li>White label</li> </ul>  |
| <b>Enrutador</b>                               | 2 disponibles para pruebas | <ul style="list-style-type: none"> <li>Ubiquiti Networks. EdgeRouter X</li> <li>5 puertos RJ45 gigabit</li> <li>Opción 'PoE passthrough' para alimentar un dispositivo desde el puerto ethernet</li> <li>Alimentación a través del adaptador o PoE pasivo 24V por el puerto ethernet</li> <li>Dual-Core 880 MHz</li> <li>256 MB DDR3 RAM.</li> </ul> |
| <b>Punto de acceso inalámbricos</b>            | 2 disponibles para pruebas | <ul style="list-style-type: none"> <li>Ubiquiti Networks. AccessPoint Outdoor+</li> <li>(2) puertos 10/100 Ethernet. 1 puerto Gigabit Ethernet compatible además con el estándar 802.3af PoE</li> </ul>  |

Tabla 1 Inventario de recursos tecnológicos de Wimix Solutions and Services

Wimix cuenta ya con suscripción mensual a Amazon Web Services S3 el cual almacena la información de la plataforma Wimix Ads, una plataforma aparte

donde se suben las imágenes a ser utilizadas en la página de publicidad de los portales cautivos. Los enrutadores y puntos de acceso se brindan a los clientes para ofrecerles el servicio de hotspots, por lo que también se utilizaron para este proyecto al momento de hacer pruebas en el sistema.

A continuación, en la Tabla 2 Requerimientos tecnológicos mínimos para la implementación del sistema), se detallan los requerimientos de recursos tecnológicos que un establecimiento cliente, al contratar el servicio de Wimix, necesita para utilizar el sistema.

| Recurso                            | Características  |
|------------------------------------|--|
| <b>Memoria RAM</b>                 | <ul style="list-style-type: none"> <li>• 500 MB</li> </ul>   |
| <b>Procesador</b>                  | <ul style="list-style-type: none"> <li>• 1 GHz</li> </ul>  |
| <b>Espacio en disco disponible</b> | <ul style="list-style-type: none"> <li>• El sistema no instala ninguna aplicación en el sistema operativo del cliente. Se necesita alrededor de 600MB para instalar un navegador web</li> </ul>  |
| <b>Sistema Operativo</b>           | <ul style="list-style-type: none"> <li>• Windows 7 o superior</li> </ul>   |
| <b>Explorador web</b>              | <ul style="list-style-type: none"> <li>• Google Chrome versión 50 o superior</li> <li>• Mozilla Firefox versión 60 o superior</li> <li>• HTML viewer Android</li> </ul>  |
| <b>Tarjeta gráfica</b>             | <ul style="list-style-type: none"> <li>• Gráficos: Tarjeta compatible con DirectX y con 64 MB*</li> </ul>  |
| <b>Otros</b>                       | <ul style="list-style-type: none"> <li>• Conexión a internet</li> </ul>  |
| <b>Enrutador</b>                   | <ul style="list-style-type: none"> <li>• Ubiquiti Networks. EdgeRouter X</li> <li>• 5 puertos RJ45 gigabit</li> <li>• Opción 'PoE passthrough' para alimentar un dispositivo desde el puerto ethernet</li> <li>• Alimentación a través del adaptador o PoE pasivo 24V por el puerto ethernet</li> <li>• Dual-Core 880 MHz <ul style="list-style-type: none"> <li>○ 256 MB DDR3 RAM.</li> </ul> </li> </ul> |
| <b>Punto de acceso</b>             | <ul style="list-style-type: none"> <li>• Ubiquiti Networks. AccessPoint Outdoor+</li> <li>• (2) puertos 10/100 Ethernet. 1 puerto Gigabit Ethernet compatible además con el estándar 802.3af PoE</li> </ul>  |

*Tabla 2 Requerimientos tecnológicos mínimos para la implementación del sistema*

## 5.1.2 Estrategia del Hardware, Software y Servicios

En esta sección se detallan los recursos de hardware necesarios para el desarrollo del sistema.

### 5.1.2.1 Hardware

En la siguiente tabla se mencionan el equipo necesario para el desarrollo del sistema, considerando que se instalarán otras herramientas para codificar y realizar pruebas.

El equipo de desarrollo hará uso de recursos propios para el desarrollo del sistema, los cuales presentan las características presentadas en la Tabla 3 Estrategia de Hardware):

| Recurso                                | Cantidad | Características  |
|--|----------|--|
| <b>Laptop</b>                          | 2        | <ul style="list-style-type: none"> <li>• Procesador x64, 2GHz o superior</li> <li>• Memoria RAM 4GB o superior</li> <li>• Espacio de Almacenamiento mínimo 160GB</li> <li>• Memoria gráfica 1 GHz</li> </ul> |
| <b>Periféricos</b>                     | 2        | <ul style="list-style-type: none"> <li>• Ratón inalámbrico</li> </ul>  |
| <b>Disco duro externo</b>              | 1        | <ul style="list-style-type: none"> <li>• 500GB de almacenamiento</li> </ul>  |
| <b>Enrutador</b>                       | 1        | <ul style="list-style-type: none"> <li>• Ubiquiti Networks. EdgeRouter X</li> </ul>  |
| <b>Punto de acceso</b>                 | 1        | <ul style="list-style-type: none"> <li>• Ubiquiti Networks. AccessPoint Outdoor+</li> </ul>  |
| <b>Enrutador con acceso a Internet</b> | 1        | <ul style="list-style-type: none"> <li>• Yota 4Gear. 1MB de internet.</li> </ul>   |

Tabla 3 Estrategia de Hardware

### 5.1.2.2 Software / Servicios

La mayoría de herramientas utilizadas para el desarrollo del sistema son de software libre y código abierto, dado al amplio auge del mismo y el aprovechamiento del internet para la búsqueda, investigación y retroalimentación sobre los mismos. Dichas herramientas son presentadas en la Tabla 4 Estrategia de Software / Servicios)

| Recurso                  | Cantidad | Características  |
|--------------------------|----------|--|
| <b>Sistema Operativo</b> | 2        | <ul style="list-style-type: none"> <li>▪ Windows 10 Pro 64bits. Microsoft Corporation</li> </ul> |



| Recurso                       | Cantidad | Características   |
|-------------------------------|----------|---|
| <b>Frameworks y Librerías</b> | 1        | <ul style="list-style-type: none"> <li>▪ GraphQL</li> <li>▪ Apollo</li> <li>▪ React.js</li> <li>▪ Ant Design</li> </ul> |
| <b>Base de Datos</b>          | 1        | <ul style="list-style-type: none"> <li>▪ MongoDB</li> </ul>   |
| <b>Servidor web/proxy</b>     | 1        | <ul style="list-style-type: none"> <li>▪ Express GraphQL server</li> <li>▪ NGINX</li> </ul>                             |
| <b>Ofimática</b>              | 1        | <ul style="list-style-type: none"> <li>▪ Microsoft Office 2016</li> </ul>   |
| <b>Navegador</b>              | 1        | <ul style="list-style-type: none"> <li>▪ Chrome versión 68 o superior</li> </ul>  |
| <b>Almacenamiento</b>         | 6 meses  | <ul style="list-style-type: none"> <li>▪ Amazon Simple Storage Service S3</li> </ul>                                    |
| <b>Dominio / Hosting</b>      | 1 año    | <ul style="list-style-type: none"> <li>▪ JustHost Plus</li> </ul>   |
| <b>Entorno de despliegue</b>  | 4 meses  | <ul style="list-style-type: none"> <li>▪ Heroku Hobby</li> </ul>  |
| <b>Otros</b>                  | 1        | <ul style="list-style-type: none"> <li>▪ GitHub</li> <li>▪ SendGrid</li> <li>▪ mLab</li> </ul>                          |

Tabla 4 Estrategia de Software / Servicios

MongoDB es el gestor de base de datos No SQL, orientado a documentos de código abierto, y mLab es un servicio en la nube para almacenar este tipo de base de datos, así como también permite hacer respaldos, analíticas y proporciona seguridad a los datos almacenados.

GraphQL no es una librería ni un framework, se trata de un lenguaje de consulta que puede ser utilizado indistintamente en diferentes códigos de programación. Es de código abierto, la comunidad hoy en día sigue creciendo y se encuentra bastante documentación sobre su uso e implementación.

Node.js y Express, se unen a GraphQL para crear el servidor web. Apollo Client y Server con ReactJS se integran para trabajar en el front-end del sistema. Y Ant Design proporciona el diseño de interfaz del sistema, siguiendo los estándares necesarios para que pueda correr en diferentes navegadores populares hoy en día.

GitHub permite almacenar y revisar el código, a esto le agregamos la ayuda de Travis CI que permite realizar pruebas al código antes de subirlo a producción y desplegarlo una vez que las pruebas sean exitosas.

Amazon Web Services S3 es una herramienta que han utilizado en Wimix a lo largo de su trayectoria, permite almacenar objetos y datos como parte de su servicio de pago mensual, en dependencia del peso de los archivos. Este servicio se ocupará para almacenar las imágenes de Wimix Ads.

### **5.1.2.3      *Equipo de Desarrollo***

El equipo de desarrollo estará conformado por 2 personas (monografistas) que se dividirán en dos funciones, uno como desarrollador front-end, el cual se encarga del desarrollo del lado del cliente, que incluye vistas, experiencia de usuario y diseño del sistema. El otro desarrollador back-end, se encargó del lado del servidor, el cual incluye conexión con la base de datos, manejo de base de datos y manejo de APIs.

Se estipuló esta división de trabajo para lograr un mayor trabajo en menor tiempo, de igual forma ambos desarrolladores se mantienen al tanto de las funciones del sistema y su estructura.

Los desarrolladores también estarán encargados de las siguientes tareas:

- Obtener y seleccionar historias de usuarios
- Dividir en tareas las historias recopiladas
- Establecer prioridades de las historias de usuario
- Estimar el tiempo de desarrollo de las historias de usuario
- Elaborar un plan de entrega
- Implementación del sistema
- Elaborar documentación

Durante la etapa de pruebas se incorporará la participación de colaboradores para ejecutar las pruebas de aceptación correspondientes a cada historia de usuario.

Los programadores deberán contar con los siguientes conocimientos presentados en la Tabla 5 Equipo de desarrollo):

| Recurso                        | Cantidad | Características   |
|--------------------------------|----------|---|
| <b>Desarrollador Front-End</b> | 1        | Conocimientos en: <ul style="list-style-type: none"> <li>• JavaScript</li> <li>• React.js</li> <li>• HTML</li> <li>• CSS</li> <li>• Experiencia de usuario</li> <li>• Interfaces de usuario</li> </ul>      |
| <b>Desarrollador Back-End</b>  | 1        | Conocimientos en: <ul style="list-style-type: none"> <li>• JavaScript - NodeJS</li> <li>• MongoDB</li> <li>• GraphQL</li> <li>• Base de datos</li> <li>• Servidores y despliegue de aplicaciones</li> </ul> |

Tabla 5 Equipo de desarrollo

## 5.2. Aspectos Económicos

A continuación, se detallan los costos de lo planteado en la factibilidad técnica para el desarrollo del sistema. La moneda utilizada para el presupuesto fue en Dólares Americanos (\$ USD) dado que esta es la moneda que Wimix utiliza para adquirir las licencias y ofrecer sus servicios.

### 5.2.1 Costo de Desarrollo

En los costos de desarrollo expuestos en la Tabla 6 Costos de desarrollo) se hace mención del pago a los desarrolladores, dichos costos se han basado en sondeos realizados en empresas funcionales y promedios de salarios en sus propuestas laborales en Nicaragua.

| Recurso                        | Cantidad | Costo (por mes) <sup>5</sup> | Duración | Total      |
|--------------------------------|----------|------------------------------|----------|------------|
| <b>Desarrollador Front-End</b> | 1        | \$ 500.00                    | 6 meses  | \$ 3000.00 |
| <b>Desarrollador Back-End</b>  | 1        | \$ 500.00                    | 6 meses  | \$ 3000.00 |
| <b>Total</b>                   |          | \$ 1000.00                   | 6 meses  | \$ 6000.00 |

Tabla 6 Costos de desarrollo

<sup>5</sup> Costos basados en salarios promedios de salarios ofrecidos en propuestas laborales en Nicaragua

### 5.2.1.1 Costos de Hardware y Software

En la Tabla 7 Costos de Hardware), se mencionan los costos de las herramientas a utilizar a lo largo del desarrollo del sistema. Se divide en Hardware, basados en cotizaciones de las partes físicas en tiendas locales. Y Software / servicios, basado en licenciamientos y servicios de terceros adquiridos de forma online.

Como se mencionó anteriormente en la factibilidad técnica, la empresa Wimix cuenta con equipos de enrutador y punto de acceso para realizar las pruebas, así también con servicios de alojamiento y dominio. Los desarrolladores cuentan con recursos tecnológicos para el desarrollo del sistema por lo que no se requerirá de una inversión inicial en la adquisición de estos recursos.

| Recurso (Hardware)        | Cantidad | Características   | Costo <sup>6</sup> |
|---------------------------|----------|---|--------------------|
| <b>Laptop</b>             | 2        | Laptop DELL INSPIRON 3579.<br><ul style="list-style-type: none"> <li>• I7-8750H 2020GHz</li> <li>• Disco Duro de 1TB</li> <li>• Disco Solido 128G</li> <li>Memoria RAM 8GB DDR4</li> <li>• Windows 10 Home Single</li> <li>• Pantalla de 15.6 Pulgadas</li> </ul> | \$ 1,344.35        |
| <b>Periféricos</b>        | 2        | Ratón Logitech M170   | \$ 12.36           |
| <b>Disco Duro Externo</b> |          | Disco duro externo 1TB<br>AHV620-1TBUSB   | \$ 65.26           |
| <b>Total</b>              |          |   | \$ 3771.71         |

Tabla 7 Costos de Hardware

Se trabajará con herramientas de código abierto y software libre, a excepción de licenciamientos y servicios especificados. En los costos expuestos en Tabla 8 Costos de Software/servicios) se refleja el precio de los servicios o productos por el tiempo de desarrollo del sistema.

Cabe aclarar que la suscripción a HotspotSystem se utilizó por 4 meses para empezar a realizar pruebas al sistema, es decir no fue utilizado a lo largo de

<sup>6</sup> Precios basados en cotización con Sevasa.

todo el desarrollo de la plataforma. Así mismo, mLab cuenta con una suscripción gratuita o Sandbox, la cual se utilizó para el desarrollo del sistema.

| Recursos  | Cant.                             | Características   | Costo <sup>7</sup> | Total     |
|---|-----------------------------------|---|--------------------|-----------|
| <b>Sistema Operativo</b>                          | 2                                 | Windows 10 Pro  | \$ 138.00          | \$ 276.00 |
| <b>Ofimática</b>                                  | 1                                 | Microsoft Office 2016   | \$ 119.99          | \$ 119.99 |
| <b>Herramientas de desarrollo</b>                 | Gratuito                          | <ul style="list-style-type: none"> <li>▪ GraphQL</li> <li>▪ Apollo</li> <li>▪ React.js</li> <li>▪ Ant Design</li> </ul> | \$ 00.00           | \$ 00.00  |
| <b>Protocolo de seguridad</b>                     | Pago anual                        | Certificado SSL   | \$ 149.00          | \$ 149.00 |
| <b>Hosting / Dominio</b>                          | Pago anual                        | JustHost  | \$ 15.99           | \$ 15.99  |
| <b>Almacenamiento de base de datos en la nube</b> | Gratuito.<br>5 meses de uso       | mLab: Sandbox   | \$ 00.00           | \$ 00.00  |
| <b>Almacenamiento de objetos y datos</b>          | Pago mensual. 6 meses de uso      | Amazon Web Service S3   | \$ 12.00           | \$ 72.00  |
| <b>Sistema de gestión de puntos de accesos</b>    | Pago mensual. 4 meses de servicio | HotspotSystem   | \$ 49.00           | \$ 196.00 |
| <b>Total</b>                                      |                                   |   | \$ 483.98          | \$ 828.98 |

*Tabla 8 Costos de Software/servicios*

### **5.2.1.2 Costos Totales**

En la siguiente tabla, Tabla 9 Costos totales) se realiza un resumen de todos los costos involucrados en el desarrollo del sistema, dando como resultado el costo estimado del proyecto.

| Recursos                          | Costo        |
|-----------------------------------|--------------|
| <b>Costos de desarrollo</b>       | \$ 6000.00   |
| <b>Costos de Hardware</b>         | \$ 3771.71   |
| <b>Costos de Software</b>         | \$ 828.98    |
| <b>Total estimado<sup>8</sup></b> | \$ 10,600.69 |

<sup>7</sup> Por unidad, pago mensual o anual.

<sup>8</sup> El total que costaría el proyecto. Cabe aclarar que, dado que es un proyecto monográfico, dicha cantidad no fue asumida totalmente por la empresa.

*Tabla 9 Costos totales*

### **5.3. Aspectos Operativos**

El desarrollo del sistema es considerado necesario para Wimix y sus clientes. Los involucrados están dispuestos a facilitar los recursos necesarios para el desarrollo eficaz del mismo; permitiendo así, que el sistema sea factible y que entre en operación sin que exista objeción alguna.

Con la finalidad de garantizar el buen funcionamiento y uso del sistema, y que impacte de forma positiva a los usuarios finales, el sistema será desarrollado con una interfaz intuitiva que permita a un usuario con conocimientos básicos en sistemas de gestión de contenido, manejarlo con facilidad.

Al implantar el sistema, facilitará y reducirá los tiempos de ejecución de las tareas correspondientes así también como disminuir la dependencia entre los clientes establecimientos y el personal de Wimix.

Los directivos de Wimix se encuentran anuentes a aceptar los cambios y mejoras que sean necesarios para que el sistema se desarrolle e implemente eficientemente, llevando a la conclusión que el sistema es factible operacionalmente.

### 5.3.2 Cronograma

| Meses   | Mes 1 |   |   |   | Mes 2 |   |   |   | Mes 2 |   |   |   | Mes 3 |   |   |   | Mes 4 |   |   |   | Mes 5 |   |   |   | Mes 6 |   |   |   | Mes 7 |   |   |   | Mes 8 |   |   |   | Mes 9 |  |  |  |
|---|-------|---|---|---|-------|---|---|---|-------|---|---|---|-------|---|---|---|-------|---|---|---|-------|---|---|---|-------|---|---|---|-------|---|---|---|-------|---|---|---|-------|--|--|--|
| Semanas   | 1     | 2 | 3 | 4 | 1     | 2 | 3 | 4 | 1     | 2 | 3 | 4 | 1     | 2 | 3 | 4 | 1     | 2 | 3 | 4 | 1     | 2 | 3 | 4 | 1     | 2 | 3 | 4 | 1     | 2 | 3 | 4 | 1     | 2 | 3 | 4 |       |  |  |  |
| Actividades   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |  |  |  |
| 1. Exploración  |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |  |  |  |
| 1.1 Entrevista con los involucrados del proyecto                          |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |  |  |  |
| 1.2 Capacitación en las historias de usuario                              |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |  |  |  |
| 1.3 Prototipos de historias de usuario                                    |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |  |  |  |
| 2. Planificación  |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |  |  |  |
| 2.1 Reunión con el tutor  |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |  |  |  |
| 2.2 Reunión con los interesados   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |  |  |  |
| 2.3 Documentación de historias de usuarios                                |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |  |  |  |
| 2.4 Planificación de Entregables  |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |  |  |  |
| 2.5 Definición de criterios de aceptación                                 |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |  |  |  |
| 3. Diseño   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |  |  |  |
| 3.1 Tarjetas CRC  |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |  |  |  |
| 3.2 Identificar las clases importantes                                    |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |  |  |  |
| 3.3 Presentación de prototipo operacional                                 |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |  |  |  |
| 3.4 Refactorización del diseño  |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |  |  |  |
| 3.5 Diseño de Base de Datos   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |  |  |  |
| 4. Codificación   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |  |  |  |
| 4.1 Desarrollo del Front-end  |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |  |  |  |
| 4.2 Desarrollo del Back-end   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |  |  |  |
| 4.3 Refactorización de Código   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |  |  |  |
| 5. Pruebas (equipo de desarrollo)   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |  |  |  |
| 5.1 Realizar pruebas unitarias  |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |  |  |  |
| 5.2 Realizar pruebas de aceptación  |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |  |  |  |
| 5.3 Realizar pruebas de integración                                       |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |  |  |  |
| 6. Producción   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |  |  |  |
| 6.1 Deploy de la plataforma   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |  |  |  |
| 6.2 Pruebas de aceptación (el cliente)                                    |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |  |  |  |
| 6.3 Corroboración de historias de usuarios con la plataforma desarrollada |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |  |  |  |
| 6.4 Capacitación del software a los involucrados                          |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |  |  |  |
| 7. Refinamiento de Documentos   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |  |  |  |
| 7.1 Revisión de documento monografico                                     |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |  |  |  |
| 7.2 Correcciones y mejoras al documento                                   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |  |  |  |
| 8. Finalización del Proyecto  |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |  |  |  |
| 8.1 Solicitud de Defensa de Tesis   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |  |  |  |
| 8.2 Pre-defensa   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |  |  |  |
| 8.9 Defensa Final   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |   |   |   |       |  |  |  |

Imagen 6 Cronograma

## **5.4. Aspectos Legales**

El desarrollo del sistema requiere de ciertos licenciamientos explicados anteriormente en la sección 5.1.5. Aspectos Económicos. Algunas licencias han sido adquiridas antes del proyecto en curso, y las demás tecnologías cuentan con licencia MTI (Licencia de Software perimitiva). Por tanto, no se estarán quebrantando leyes que hagan a este sistema sensible a fraudes o inconvenientes legales. Dicho de otra manera, el desarrollo de este sistema se considera viable legalmente.

A continuación, se presenta la Ley de Protección de datos personales, que todos los sitios que requieran de información sobre usuarios necesitan implementar:

### **LEY DE PROTECCIÓN DE DATOS PERSONALES**

**Ley No. 787, Aprobada el 21 de marzo del 2012**

**Publicada en La Gaceta No. 61 del 29 de marzo del 2012**

Dicha ley tiene por objeto la protección de la persona natural o jurídica frente al tratamiento, automatizado o no, de sus datos personales en ficheros de datos públicos y privados, a efecto de garantizar el derecho a la privacidad personal y familiar y el derecho a la autodeterminación informativa.



# Fase de Exploración

## **VI. APLICACIÓN DE LA METODOLOGÍA XP**

Para el desarrollo de esta monografía se eligió la metodología XP considerando que permite el trabajo el equipo, incluyendo al cliente mismo en las actividades de escritura de historias de usuario. Además de que permite pequeños lanzamientos que van siendo aprobados a medida que se va terminado.

A continuación, se describen las fases utilizadas a lo largo del proyecto.

### **6.1. Fase de Exploración**

Durante la fase de exploración se incluye el modelo del negocio, así mismo se capacitó al personal involucrado en la elaboración de las historias de usuario y metodología a utilizar. Las entrevistas y reuniones fueron parte del proceso para obtener resultados en esta primera fase.

## 6.1.2 Modelo del Negocio

Como primera etapa del análisis del modelo de negocio se presenta un diagrama con las funciones correspondiente a cada área del negocio:

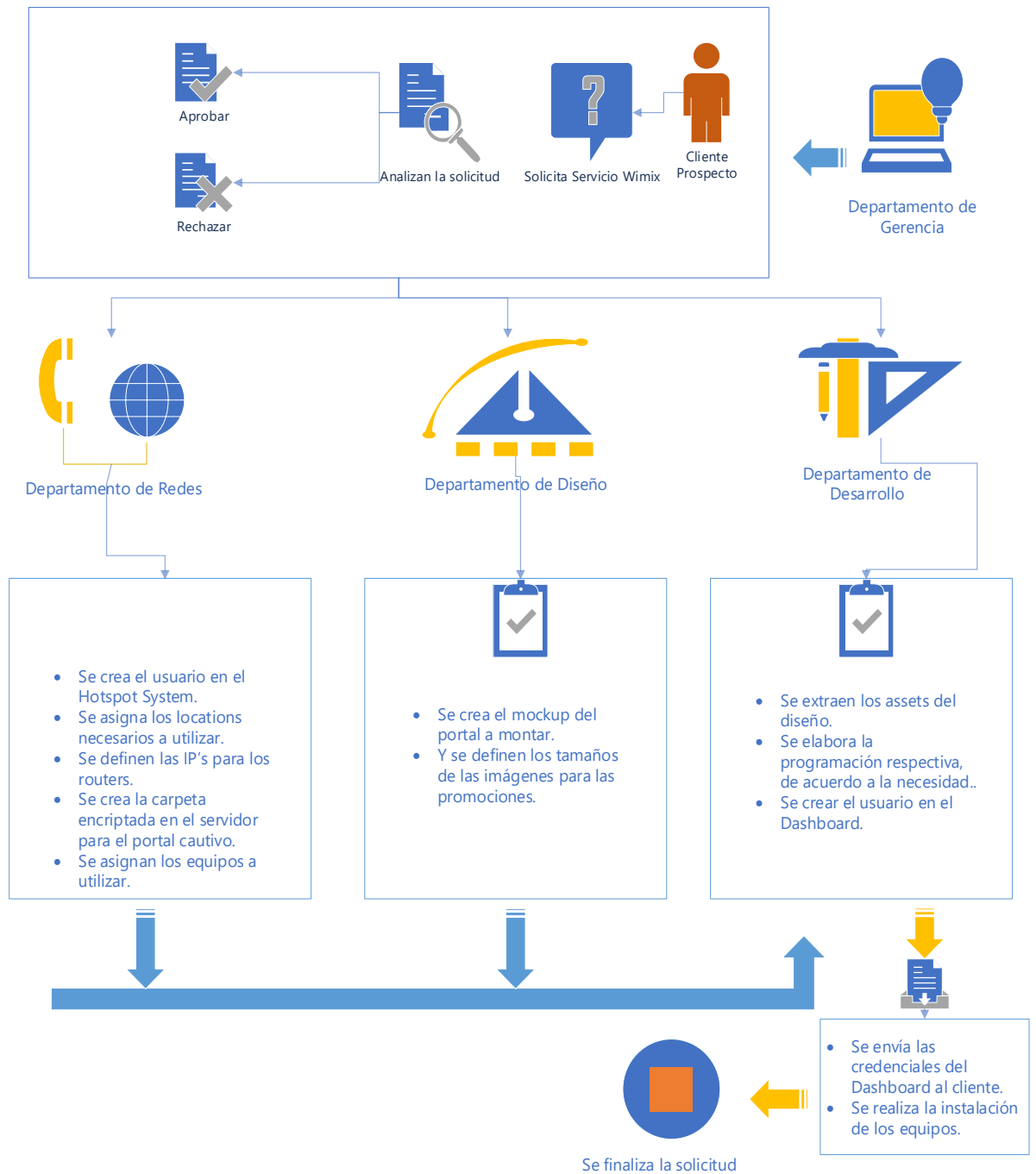


Imagen 7 Modelo del negocio

## 6.1.4 Requerimientos del sistema

### 6.1.4.1 *Requerimientos Funcionales*

El sistema permitirá:

- **Usuario Establecimiento Cliente y Usuario Wimix**
  - Crear, editar y eliminar los campos de preguntas que los establecimientos clientes requieran, y pre visualizar el formulario antes de guardarlo.
  - Crear, editar y eliminar los campos del formulario de registro de usuarios finales, y pre visualizar el formulario antes de guardarlo.
  - Configurar una página informativa, donde podrá mostrar un título y un párrafo.
  - Editar la forma en que los usuarios finales visualicen las promociones.
  - Ver una lista con los dispositivos conectados.
  - Ver una lista de todos los usuarios dispositivos registrados con filtros según convenga.
  - Ver el historial completo de conexiones hechas ordenadas por sucursales.
  - Obtener las respuestas de los usuarios a las encuestas realizadas.
  - Generar un histórico de inicios y cierres de sesión de usuarios al sistema.
  - El sistema proporcionará plantillas predeterminadas de los distintos formatos requeridos.
- **Usuario Wimix**
  - Diferentes roles y permisos.
  - Iniciar sesión con nombre de usuario y contraseña.
  - Configuración general del sistema previo a la entrega a los establecimientos cliente (Wimix Ads, locations, sucursales)

- **Usuario general**

- Visualizar y exportar resultados de las encuestas.
- Visualizar y exportar datos de todos los dispositivos que se hayan conectado.

#### **6.1.4.2      *Requerimientos no funcionales***

El sistema permitirá:

- El acceso desde equipos conectados a Internet, desde diferentes navegadores
- El acceso multiusuario
- La autenticación de usuarios por medio de un nombre de usuario y contraseña
- Proporcionar un entorno visual sencillo, de fácil interacción con el usuario.

#### **6.1.5 Historias Usuarios**

Las Historias de Usuarios tienen el mismo propósito que los casos de uso y constituyen una técnica utilizada en el desarrollo de proyectos XP, las mismas que permiten especificar los requerimientos de usuario necesarios para el desarrollo de la aplicación, mediante la utilización de tarjetas en las cuales se describe las características que el sistema debe poseer. La identificación de las historias de usuario depende de la complejidad del sistema y son descompuestas en tareas que son asignadas al programador para ser implementadas durante una iteración.

##### **6.1.5.1      *Especificación de las historias de usuario***

- **Iteración A**

| Historia de Usuario                                       |  |
|---|--|
| <b>Número:</b> HU-01-A                                    | <b>Usuario:</b> Administrador de establecimiento cliente |
| <b>Nombre historia:</b> Gestión de formulario de registro |  |
| <b>Modificación de Historia Número:</b> NA                | <b>Iteración asignada:</b> A                             |
| <b>Prioridad en negocio:</b> Alta                         |  |

| Historia de Usuario  |
|--|
| <p><b>Descripción:</b> Deseo que el administrador del sistema pueda seleccionar una plantilla de formularios de registro, desde donde sus clientes podrán registrarse para luego tener acceso a internet. Además, podrá pre visualizar el formulario para tener una idea de cómo lucirá antes de guardar los cambios.</p>  |
| <p><b>Criterios de aceptación</b></p> <ul style="list-style-type: none"> <li>• Delimitar si serán campos de texto que solo acepten números, correos y letras.</li> <li>• El usuario podrá seleccionar la plantilla de registro que desee y requiera o bien podrá crear su propia plantilla con los campos de la lista blanca.</li> <li>• Las distintas plantillas podrán contener los siguientes datos: <ul style="list-style-type: none"> <li>○ Nombre</li> <li>○ Teléfono</li> <li>○ Correo</li> <li>○ Género</li> <li>○ Cédula</li> <li>○ Fecha de nacimiento</li> <li>○ Imagen de logo de empresa</li> </ul> </li> </ul> |

Tabla 10 Historia de Usuario – Gestión de formulario de registro

| Historia de Usuario   |   |
|---|---|
| Número: HU-02-A   | Usuario: Administrador de establecimiento cliente |
| Nombre historia: Gestión de formulario de encuestas   |   |
| Modificación de Historia Número: NA   | Iteración asignada: A                             |
| Prioridad en negocio: Alta  |   |
| Descripción: El usuario administrador requiere crear encuestas para que sus clientes puedan llenarlas antes de obtener acceso a internet. Por tanto, el sistema permitirá editar y pre visualizar los formularios de encuestas. También, habrá una sección donde se puedan ver las respuestas que los clientes ingresaron en los campos de la misma.            |   |
| Criterios de aceptación   |   |
| <ul style="list-style-type: none"><li>• El usuario podrá configurar períodos de tiempo para recolectar las respuestas de los clientes.</li><li>• Existirán plantillas predeterminadas para encuestas.</li><li>• Validación de los campos de texto (que se pueda seleccionar si el campo permitirá únicamente letras, números o caracteres especiales)</li></ul> |   |

Tabla 11 Historia de usuario - Gestión de formulario de encuestas

| Historia de Usuario                 |                              |
|-------------------------------------|------------------------------|
| Número: HU-03-A                     | Usuario: Administrador Wimix |
| Nombre historia: Ajustes generales  |                              |
| Modificación de Historia Número: NA | Iteración asignada: A        |
| Prioridad en negocio: Alta          |                              |

| Historia de Usuario  |
|--|
| <b>Descripción:</b> Se podrá crear usuarios nuevos y gestionar establecimientos clientes.  |
| <b>Criterios de aceptación</b> <ul style="list-style-type: none"> <li>• Que pueda ser habilitada o deshabilitada según convenga o determine el usuario.</li> <li>• El usuario podrá subir sus propias imágenes.</li> <li>• Validar que el límite de tamaño y peso de la imagen sea el establecido por el sistema.</li> <li>• El usuario tendrá a disposición plantillas pre determinadas para seleccionar la que desee.</li> </ul> |

Tabla 12 Historia de usuario - Configuraciones generales del sistema

| Historia de Usuario  |   |
|--|---|
| Número: HU-04-A  | Usuario: Administrador de establecimiento cliente |
| Nombre historia: Autenticación y roles de usuario  |   |
| Modificación de Historia Número: NA  | Iteración asignada: A                             |
| Prioridad en negocio: Alta   |   |
| Descripción: Como usuario del sistema, deseo ingresar a la plataforma utilizando mi usuario y contraseña.  |   |
| Criterios de aceptación:   |   |
| <ul style="list-style-type: none"><li>• Que no sea posible la opción de crear un usuario/cuenta.</li><li>• Establecer diferentes roles para que puedan visualizar lo correspondiente a sus labores: root, administrador, general</li></ul> |   |

Tabla 13 Historia de usuario - Autenticación y roles de usuario

## • Iteración B

| Historia de Usuario   |   |
|---|---|
| Número: HU-01-B   | Usuario: Administrador de establecimiento cliente |
| Nombre historia: Conexiones recientes   |   |
| Modificación de Historia Número: NA   | Iteración asignada: B                             |
| Prioridad en negocio: Media   |   |
| Descripción: Deseo que el administrador del sistema de los establecimientos clientes puedan ver los dispositivos conectados al hotspot wimix para monitorear las sesiones activas.                                |   |
| Criterios de aceptación: <ul style="list-style-type: none"><li>Las sesiones mostradas serán las de los últimos de 15min</li><li>Se refrescará automáticamente</li><li>Deben existir filtros de búsqueda</li></ul> |   |

Tabla 14 Historia de usuario - Conexiones recientes

| Historia de Usuario   |  |
|---|--|
| <b>Número:</b> HU-02-B  | <b>Usuario:</b> Administrador de establecimiento cliente |
| <b>Nombre historia:</b> Administración de promociones   |  |
| <b>Modificación de Historia Número:</b> NA  | <b>Iteración asignada:</b> B                             |
| <b>Prioridad en negocio:</b> Media  |  |
| <b>Descripción:</b> El sistema debe permitir que el usuario del establecimiento pueda cambiar la forma en que sus clientes ven las promociones. Esto puede ser de dos opciones: en vertical u horizontal. |  |

*Tabla 15 Historia de usuario - Administración de promociones*

| Historia de Usuario   |  |
|---|--|
| <b>Número:</b> HU-03-B  | <b>Usuario:</b> Administrador de establecimiento cliente |
| <b>Nombre historia:</b> Visualización de usuarios registrados   |  |
| <b>Modificación de Historia Número:</b> NA  | <b>Iteración asignada:</b> B                             |
| <b>Prioridad en negocio:</b> Media  |  |
| <b>Descripción:</b> Se requiere una lista con los usuarios finales que se han registrado desde el primer día que activaron el servicio con el fin de tener un historial de todos sus clientes y que sea aprovechado para su estrategia de ventas y marca. |  |

*Tabla 16 Historia de usuario - Visualización de usuarios registrados*

- **Iteración C**

| Historia de Usuario   |  |
|---|--|
| <b>Número:</b> HU-01-C  | <b>Usuario:</b> Administrador de establecimientos clientes |
| <b>Nombre historia:</b> Panel principal / Dashboard   |  |
| <b>Modificación de Historia Número:</b> NA  | <b>Iteración asignada:</b> C                               |
| <b>Prioridad en negocio:</b> Baja   |  |
| <b>Descripción:</b> El sistema deberá mostrar un panel de control principal o escritorio donde se ubicarán widgets con estadísticas básicas de las conexiones recientes. También, el sistema generará un código de conexión diario con el cual sus clientes podrán ingresar directamente al hotspot, notificaciones de correos enviados, notificaciones de nuevas respuestas a encuestas. |  |

*Tabla 17 Historia de usuario - Panel principal / Dashboard*

| Historia de Usuario                             |  |
|---|--|
| <b>Número:</b> HU-02-C                          | <b>Usuario:</b> Administrador de establecimiento cliente |
| <b>Nombre historia:</b> Historial de conexiones |  |
| <b>Modificación de Historia Número:</b> NA      | <b>Iteración asignada:</b> C                             |
| <b>Prioridad en negocio:</b> Baja               |  |



| Historia de Usuario  |  |
|--|--|
| <b>Descripción:</b> Quiero que el administrador del sistema pueda ver un listado a forma de historial de todas las conexiones de los clientes hechos al hotspot Wimix desde el primer día en el que se activó el servicio hasta la actualidad, con el fin de tener un histórico de las sesiones generadas y que ésta pueda ser de utilidad para su estrategia de ventas y marca. Además, se podrán filtrar según convenga. |  |

*Tabla 18 Historia de usuario - Historial de conexiones*

| Historia de Usuario  |  |
|--|--|
| <b>Número:</b> HU-03-C   | <b>Usuario:</b> Administrador de establecimiento cliente |
| <b>Nombre historia:</b> Bitácora   |  |
| <b>Modificación de Historia Número:</b> NA   | <b>Iteración asignada:</b> C                             |
| <b>Prioridad en negocio:</b> Baja  |  |
| <b>Descripción:</b> Se necesita que el sistema muestre los inicios y cierres de sesión de usuarios en el sistema |  |

*Tabla 19 Historia de usuario - Bitácora*

| Historia de Usuario  |  |
|--|--|
| <b>Número:</b> HU-03-D   | <b>Usuario:</b> Administrador de establecimiento cliente |
| <b>Nombre historia:</b> Autenticidad de correos  |  |
| <b>Modificación de Historia Número:</b> NA   | <b>Iteración asignada:</b> C                             |
| <b>Prioridad en negocio:</b> Media   |  |
| <b>Descripción:</b> Se requiere que el sistema verifique si el correo ingresado por el usuario al registrarse es válido y auténtico. |  |

*Tabla 20 Historia de usuario - Autenticidad de correos*

# Capítulo II:

# Fase de Planificación

## 6.2. Fase de Planificación

Establecimiento de prioridades de las historias de usuario.

El propósito de establecer prioridades es para evaluar la importancia de cada historia, en cuanto a su contribución o aporte al desarrollo del sistema; y de acuerdo a las especificaciones establecidas por el cliente. Las mismas, fueron evaluadas y aprobadas por Wimix.

| Iteración | Historia de usuario |                                       | Prioridad | Dependencia   |
|-----------|---------------------|---------------------------------------|-----------|---|
|           | Número              | Nombre                                |           |   |
| <b>A</b>  | HU-01-A             | Gestión de formulario de registro     | Alta      | N/A   |
|           | HU-02-A             | Gestión de formulario de encuestas    | Alta      | N/A   |
|           | HU-03-A             | Ajustes generales                     | Alta      | N/A   |
|           | HU-04-A             | Autenticación                         | Alta      | N/A   |
| <b>B</b>  | HU-01-B             | Conexiones Recientes                  | Media     | N/A   |
|           | HU-02-B             | Administración de promociones         | Media     | N/A   |
|           | HU-03-B             | Visualización de usuarios registrados | Media     | HU-01-A   |
| <b>C</b>  | HU-01-C             | Panel principal / Dashboard           | Media     | HU-01-A<br>HU-02-A<br>HU-03-A<br>HU-01-B<br>HU-03-B |
|           | HU-02-C             | Historial de conexiones               | Baja      | HU-01-A<br>HU-01-B                                  |
|           | HU-03-C             | Bitácora                              | Baja      | HU-04-A   |

Tabla 21 Establecimiento de prioridades de las HS

Los aspectos considerados de Alta prioridad, se establecieron en la iteración A que será la primera; fueron estimados para la primera fase dado que se pretende disminuir la dependencia entre Wimix y los establecimientos clientes.

La siguiente iteración B, cuenta con historias de Media prioridad. Los cuales dependen de otras historias de usuario o bien son consideradas imprescindibles para una primera entrega.

Finalmente, la iteración C cuenta con historias de Media – Baja prioridad. Las historias dentro de esta iteración dependen de otras y con ellas se concluiría un producto base.

## 6.2.1 Cronograma de tareas

- Iteración A

| Nombre de la tarea   | Duración       | Inicio            | Final             |
|--|----------------|-------------------|-------------------|
| <b>HU-01-A Gestión de formulario de registro</b>   | <b>12 días</b> | <b>1/10/2018</b>  | <b>16/10/2018</b> |
| Crear las colecciones respectivas en la BD   | 1 día          | 1/10/2018         | 1/10/2018         |
| Crear el layout de los campos que puede solicitar  | 3 días         | 1/10/2018         | 3/10/2018         |
| Crear los componentes necesarios que permitan la personalización de los formularios de registro  | 3 días         | 3/10/2018         | 5/10/2018         |
| Crear la página de previsualización del formulario creado/seleccionado                           | 3 días         | 8/10/2018         | 10/10/2018        |
| Validar los campos con los tipos de datos correspondientes                                       | 1 día          | 11/10/2018        | 11/10/2018        |
| Realizar las pruebas de funcionalidad de las interfaces  | 1 día          | 11/10/2018        | 11/10/2018        |
| Refactorización  | 1 día          | 12/10/2018        | 12/10/2018        |
| Corrección de errores  | 3 días         | 12/10/2018        | 16/10/2018        |
| Actualizar la documentación  | 1 día          | 16/10/2018        | 16/10/2018        |
| <b>HU-02-A Gestión de formulario de encuestas</b>  | <b>12 días</b> | <b>17/10/2018</b> | <b>1/11/2018</b>  |
| Crear las colecciones respectivas en la BD   | 1 día          | 17/10/2018        | 17/10/2018        |
| Crear el layout de los campos que puede solicitar  | 2 días         | 18/10/2018        | 19/10/2018        |
| Crear los componentes necesarios que permitan la personalización de los formularios de encuestas | 1 día          | 22/10/2018        | 22/10/2018        |
| Crear la página de previsualización del formulario creado/seleccionado                           | 3 días         | 23/10/2018        | 25/10/2018        |
| Crear un panel de visualización de las respuestas a las encuestas                                | 1 día          | 26/10/2018        | 26/10/2018        |
| Realizar las pruebas de funcionalidad de las interfaces  | 1 día          | 29/10/2018        | 29/10/2018        |
| Refactorización  | 1 día          | 29/10/2018        | 29/10/2018        |
| Corrección de errores  | 2 días         | 30/10/2018        | 31/10/2018        |
| Actualizar la documentación  | 1 día          | 1/11/2018         | 1/11/2018         |
| <b>HU-03-A Ajustes generales</b>   | <b>13 días</b> | <b>2/11/2018</b>  | <b>20/11/2018</b> |
| Crear las colecciones respectivas en la BD   | 1 día          | 2/11/2018         | 2/11/2018         |
| Crear los componentes necesarios para las interfaces   | 4 días         | 2/11/2018         | 7/11/2018         |
| Crear un panel de visualización de los usuarios y sus roles                                      | 6 días         | 8/11/2018         | 15/11/2018        |
| Realizar las pruebas de funcionalidad de las interfaces  | 3 días         | 15/11/2018        | 19/11/2018        |
| Refactorización  | 2 días         | 19/11/2018        | 20/11/2018        |
| Corrección de errores  | 2 días         | 19/11/2018        | 20/11/2018        |
| Actualizar la documentación  | 1 día          | 20/11/2018        | 20/11/2018        |
| <b>HU-04-A Autenticación</b>   | <b>6 días</b>  | <b>21/11/2018</b> | <b>29/11/2018</b> |
| Crear las colecciones respectivas en la BD   | 1 día          | 21/11/2018        | 21/11/2018        |
| Crear los componentes necesarios para las interfaces   | 4 días         | 21/11/2018        | 26/11/2018        |
| Realizar las pruebas de funcionalidad de las interfaces  | 2 días         | 26/11/2018        | 27/11/2018        |
| Refactorización  | 1 día          | 27/11/2018        | 27/11/2018        |
| Corrección de errores  | 3 días         | 27/11/2018        | 29/11/2018        |
| Actualizar la documentación  | 1 día          | 29/11/2018        | 29/11/2018        |

Imagen 8 Cronograma de tareas. Iteración A

- Iteración B

|   |                |                   |                   |
|---|----------------|-------------------|-------------------|
| <b>HU-01-B Conexiones Recientes</b>                     | <b>19 días</b> | <b>30/11/2018</b> | <b>28/12/2018</b> |
| Crear las colecciones respectivas en la BD              | 1 día          | 30/11/2018        | 30/11/2018        |
| Crear los componentes necesarios para las interfaces    | 6 días         | 3/12/2018         | 10/12/2018        |
| Crear un panel de visualización                         | 8 días         | 11/12/2018        | 20/12/2018        |
| Realizar las pruebas de funcionalidad de las interfaces | 3 días         | 20/12/2018        | 24/12/2018        |
| Refactorización   | 2 días         | 25/12/2018        | 26/12/2018        |
| Corrección de errores                                   | 1 día          | 27/12/2018        | 27/12/2018        |
| Actualizar la documentación                             | 1 día          | 28/12/2018        | 28/12/2018        |
| <b>HU-02-B Administración de promociones</b>            | <b>11 días</b> | <b>28/12/2018</b> | <b>14/1/2019</b>  |
| Crear las colecciones respectivas en la BD              | 1 día          | 28/12/2018        | 28/12/2018        |
| Crear los componentes necesarios para las interfaces    | 2 días         | 2/1/2019          | 4/1/2019          |
| Crear un panel de visualización                         | 4 días         | 4/1/2019          | 8/1/2019          |
| Realizar las pruebas de funcionalidad de las interfaces | 1 día          | 8/1/2019          | 9/1/2019          |
| Refactorización   | 1 día          | 10/1/2019         | 11/1/2019         |
| Corrección de errores                                   | 1 día          | 14/1/2019         | 14/1/2019         |
| Actualizar la documentación                             | 1 día          | 14/1/2019         | 14/1/2019         |
| <b>HU-03-B Visualización de usuarios registrados</b>    | <b>12 días</b> | <b>15/1/2019</b>  | <b>31/1/2019</b>  |
| Crear las colecciones respectivas en la BD              | 1 día          | 15/1/2019         | 15/1/2019         |
| Crear los componentes necesarios para las interfaces    | 4 días         | 15/1/2019         | 18/1/2019         |
| Crear un panel de visualización de los usuarios         | 5 días         | 21/1/2019         | 25/1/2019         |
| Realizar las pruebas de funcionalidad de las interfaces | 2 días         | 28/1/2019         | 29/1/2019         |
| Refactorización   | 2 días         | 29/1/2019         | 30/1/2019         |
| Corrección de errores                                   | 1 día          | 31/1/2019         | 31/1/2019         |
| Actualizar la documentación                             | 1 día          | 31/1/2019         | 31/1/2019         |

Imagen 9 Cronograma de tareas. Iteración B

- Iteración C

|   |                |                  |                  |
|---|----------------|------------------|------------------|
| <b>HU-01-C Panel principal / Dashboard</b>                  | <b>20 días</b> | <b>1/2/2019</b>  | <b>28/2/2019</b> |
| Crear las colecciones respectivas en la BD                  | 1 día          | 1/2/2019         | 1/2/2019         |
| Crear los componentes necesarios para las interfaces        | 9 días         | 4/2/2019         | 14/2/2019        |
| Crear un panel de visualización de los usuarios y sus roles | 4 días         | 15/2/2019        | 20/2/2019        |
| Realizar las pruebas de funcionalidad de las interfaces     | 2 días         | 21/2/2019        | 22/2/2019        |
| Refactorización   | 2 días         | 25/2/2019        | 26/2/2019        |
| Corrección de errores                                       | 1 día          | 27/2/2019        | 27/2/2019        |
| Actualizar la documentación                                 | 1 día          | 28/2/2019        | 28/2/2019        |
| <b>HU-02-C Historial de conexiones</b>                      | <b>15 días</b> | <b>1/3/2019</b>  | <b>21/3/2019</b> |
| Crear las colecciones respectivas en la BD                  | 1 día          | 1/3/2019         | 1/3/2019         |
| Crear los componentes necesarios para las interfaces        | 5 días         | 4/3/2019         | 8/3/2019         |
| Crear un panel de visualización de los usuarios y sus roles | 5 días         | 11/3/2019        | 15/3/2019        |
| Realizar las pruebas de funcionalidad de las interfaces     | 2 días         | 18/3/2019        | 19/3/2019        |
| Refactorización   | 1 día          | 20/3/2019        | 20/3/2019        |
| Corrección de errores                                       | 1 día          | 21/3/2019        | 21/3/2019        |
| Actualizar la documentación                                 | 1 día          | 21/3/2019        | 21/3/2019        |
| <b>HU-03-C Bitácora</b>                                     | <b>10 días</b> | <b>21/3/2019</b> | <b>21/3/2019</b> |
| Crear las colecciones respectivas en la BD                  | 1 día          | 22/3/2019        | 22/3/2019        |
| Crear los componentes necesarios para las interfaces        | 4 días         | 25/3/2019        | 28/3/2019        |
| Crear un panel de visualización de los usuarios y sus roles | 4 días         | 29/3/2019        | 3/4/2019         |
| Realizar las pruebas de funcionalidad de las interfaces     | 1 día          | 4/4/2019         | 4/4/2019         |
| Refactorización   | 1 día          | 4/4/2019         | 5/4/2019         |
| Corrección de errores                                       | 1 día          | 4/4/2019         | 5/4/2019         |
| Actualizar la documentación                                 | 1 día          | 5/4/2019         | 5/4/2019         |

*Imagen 10 Cronograma de tareas. Iteración C*

# Fase de Diseño



## 6.3. Fase de Diseño

### 6.3.1 Diagramas de actividades

El diagrama de actividades muestra un proceso de negocio o un proceso de software como un flujo de trabajo a través de una serie de acciones. Las personas, los componentes de software o los equipos pueden realizar estas acciones. A continuación, se muestran los diagramas de actividades para las acciones en el sistema:

- Diagrama de actividades – Autenticación

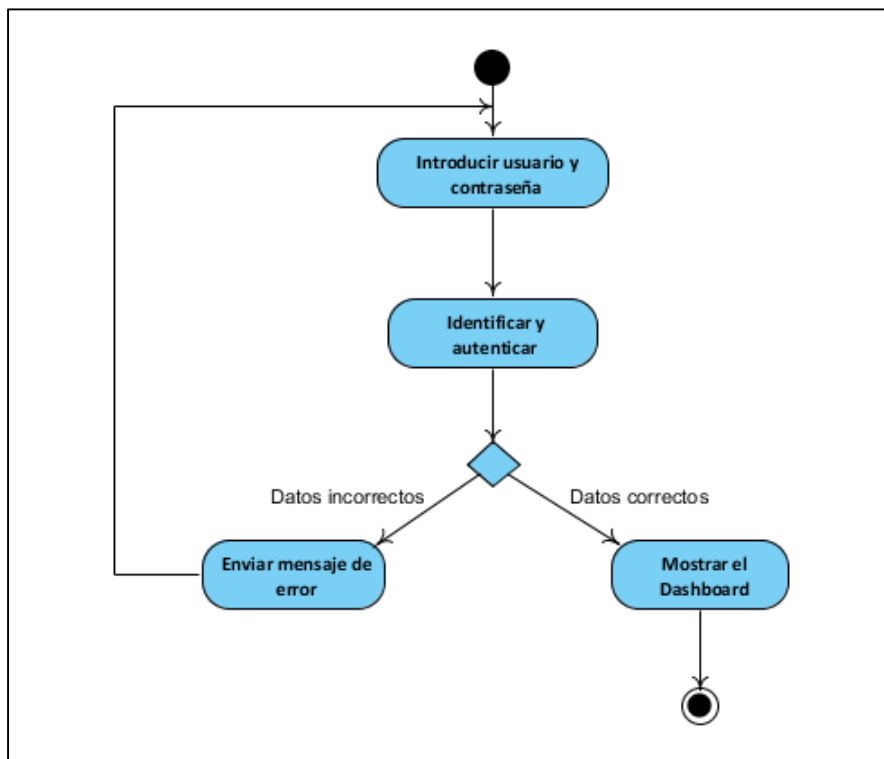


Imagen 11 Diagrama de actividades - Autenticación



- Diagrama de actividades - Gestión de formulario de registro

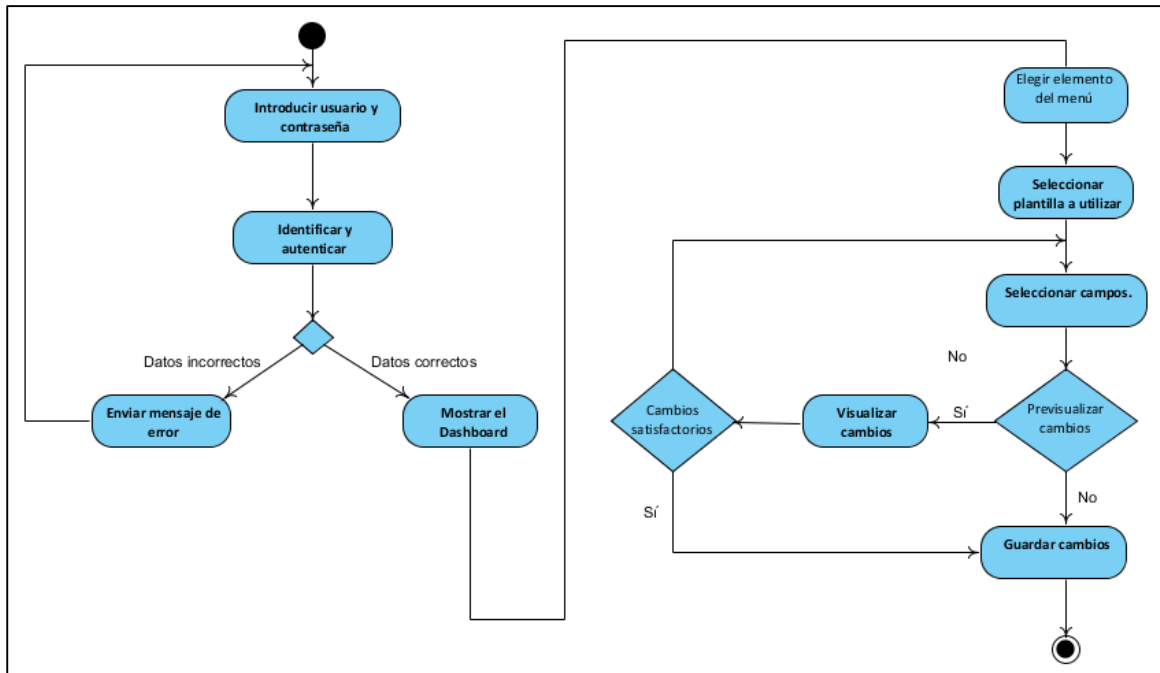


Imagen 12 Diagrama de actividades - Gestión de formulario de registro

- Diagrama de actividades - Gestión de formulario de encuestas

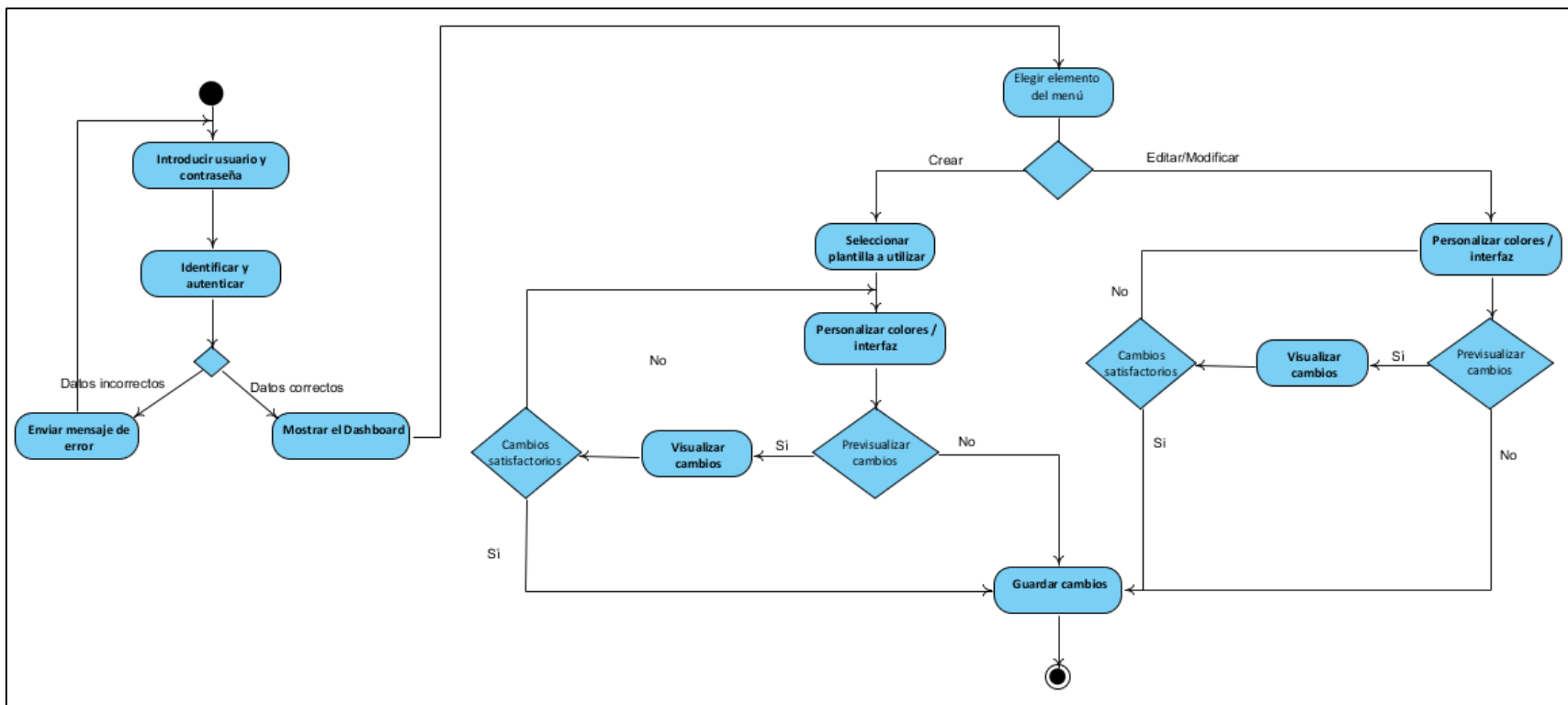


Imagen 13 Diagrama de actividades - Gestión de formulario de encuestas

- Diagrama de actividades – Conexiones Recientes

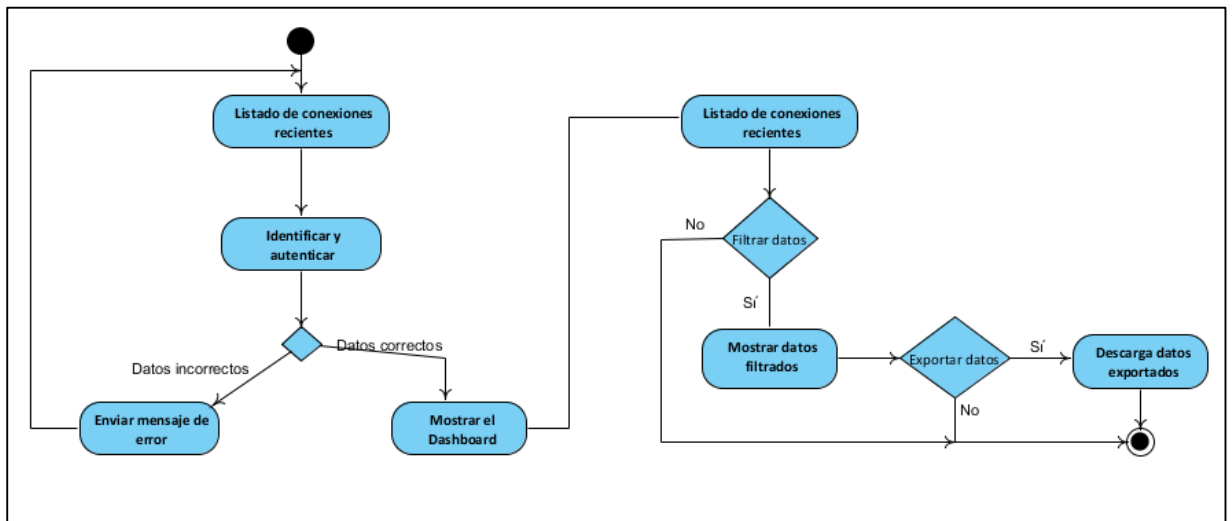


Imagen 14 Diagrama de actividades – Conexiones Recientes

- Diagrama de actividades – Administración de promociones

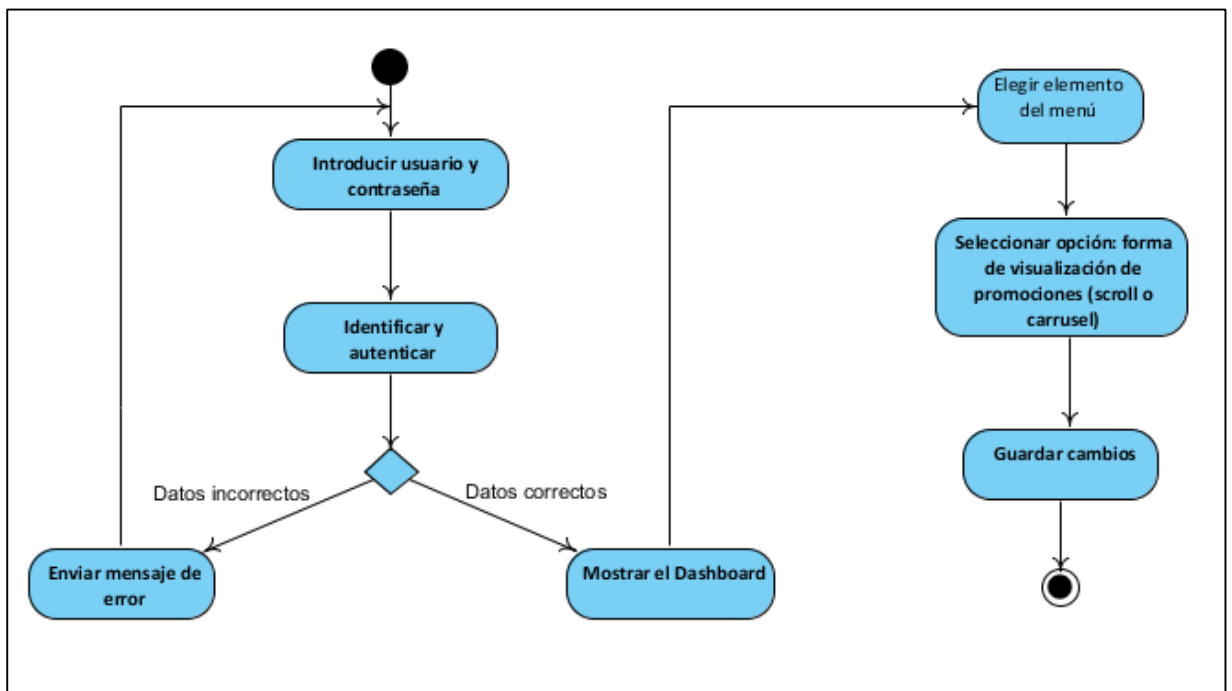


Imagen 15 Diagrama de actividades - Administración de promociones

- Diagrama de actividades – Visualización de dispositivos registrados

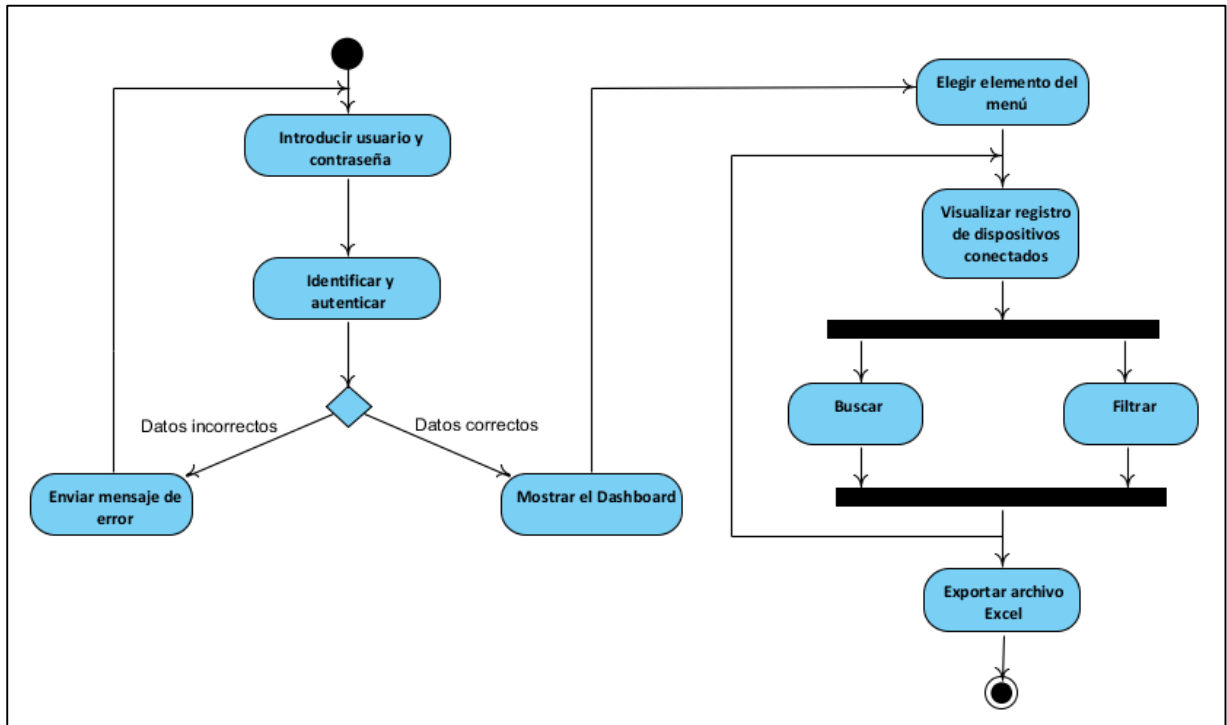


Imagen 16 Diagrama de actividades - Visualización de dispositivos registrados

- Diagrama de actividades – Panel principal / Dashboard

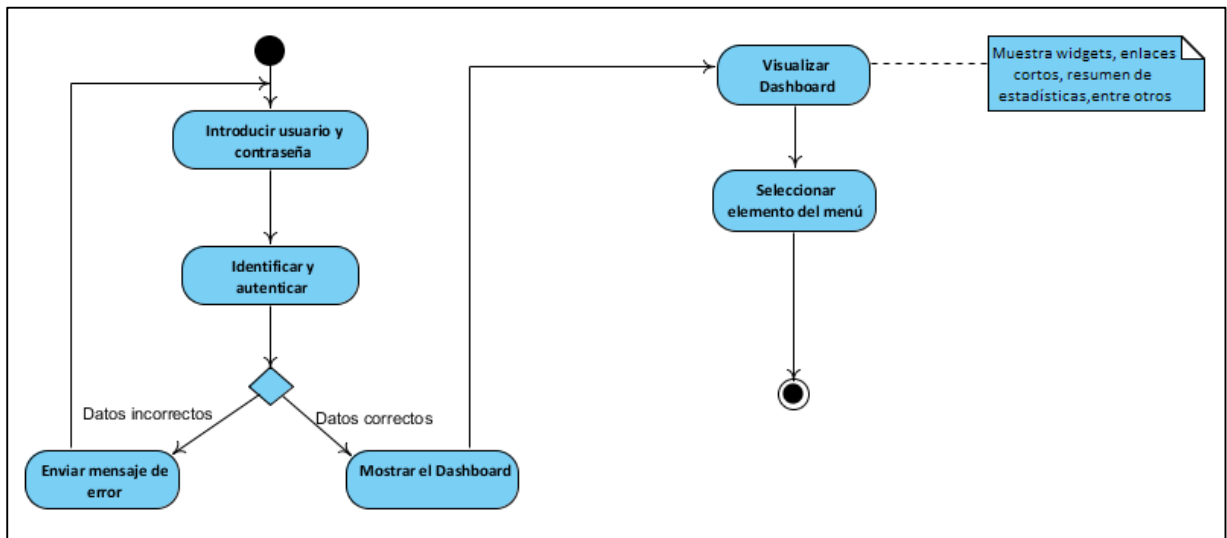


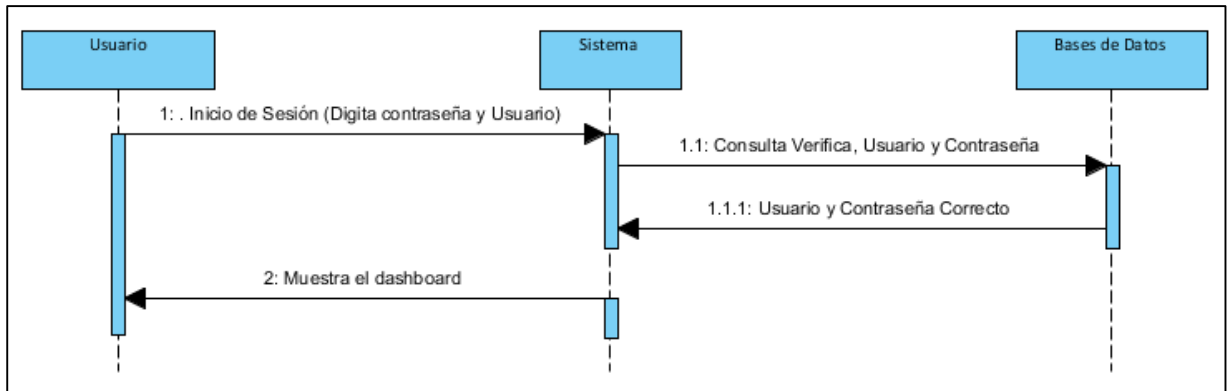
Imagen 17 Diagrama de actividades – Panel principal / Dashboard

### 6.3.2 Diagramas de secuencia

Los diagramas de secuencia muestran una interacción, que representa la secuencia de mensajes entre instancias de clases, componentes, subsistemas o

actores. El tiempo fluye por el diagrama y muestra el flujo de control de un participante a otro.

- Diagrama de secuencia – Autenticación



*Imagen 18 Diagrama de secuencia - Autenticación*

- Diagrama de secuencia - Gestión de formulario de registro

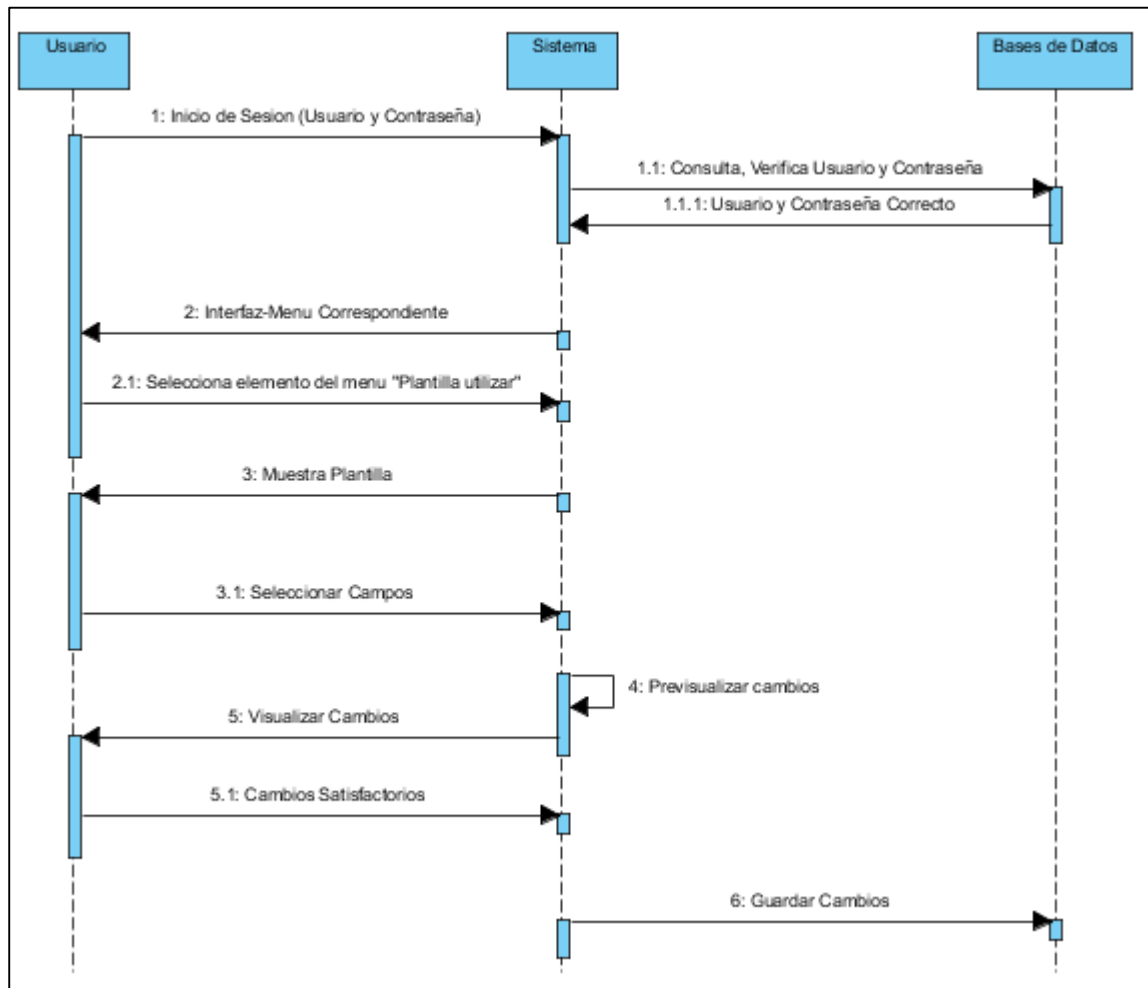
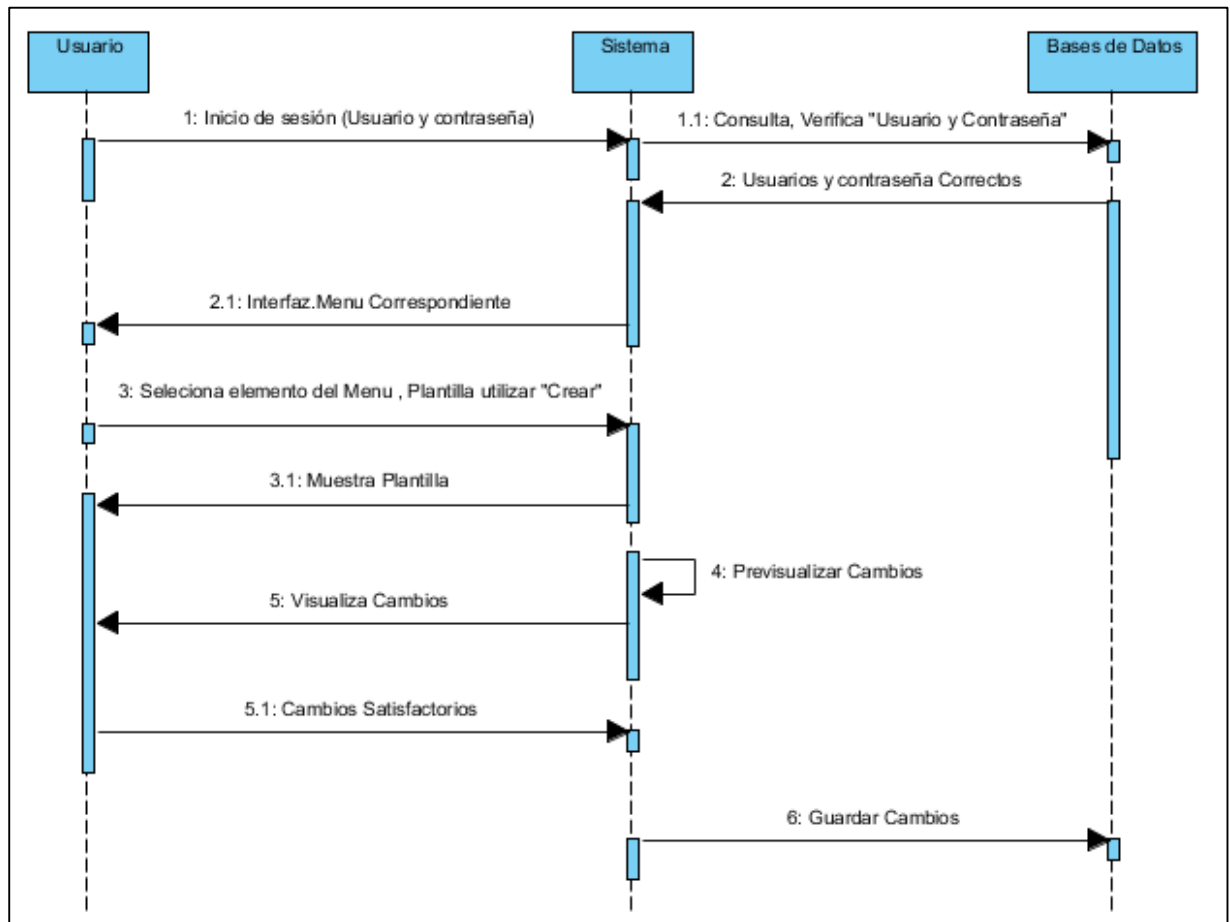


Imagen 19 Diagrama de secuencia - Gestión de formulario de registro

- Diagrama de secuencia - Gestión de formulario de encuestas



*Imagen 20 Diagrama de secuencia - Gestión de formulario de encuestas*

- Diagrama de secuencia – Conexiones Recientes

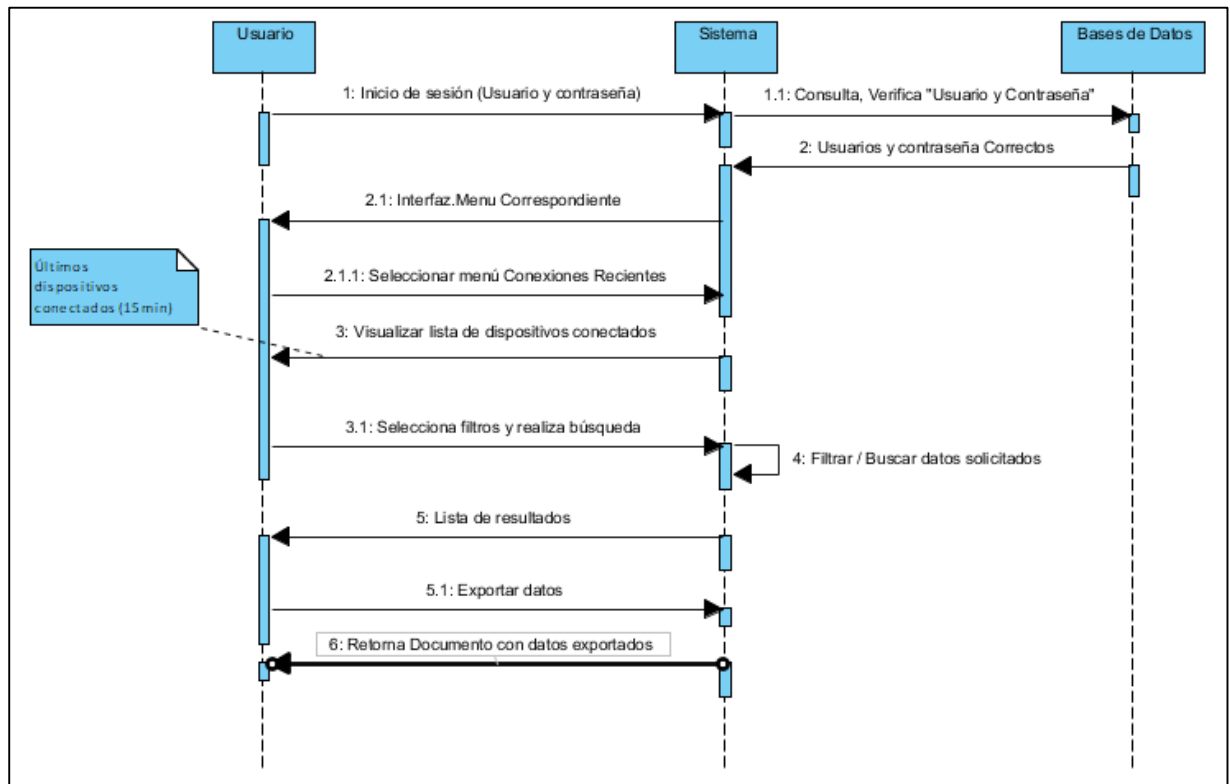


Imagen 21 Diagrama de secuencia – Conexiones Recientes



### 6.3.3 Diagramas de navegación

Los diagramas de navegación muestran una estructura de los contenidos que van a aparecer en el sitio web o aplicación. Permiten conocer el orden de flujo que lleva el sistema y presentar enlaces para comprobar la accesibilidad de los mismos en el producto.

- Diagrama de navegación para usuario administrador

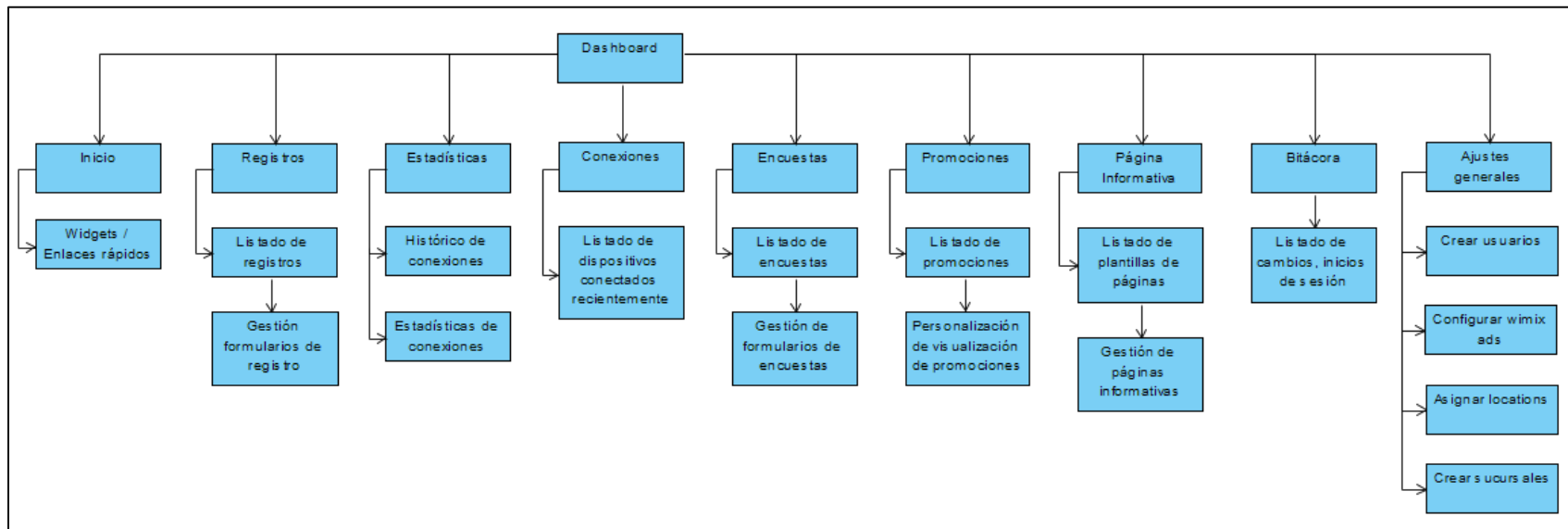


Imagen 22 Diagrama de navegación para usuario administrador

- Diagrama de navegación para usuario establecimiento cliente

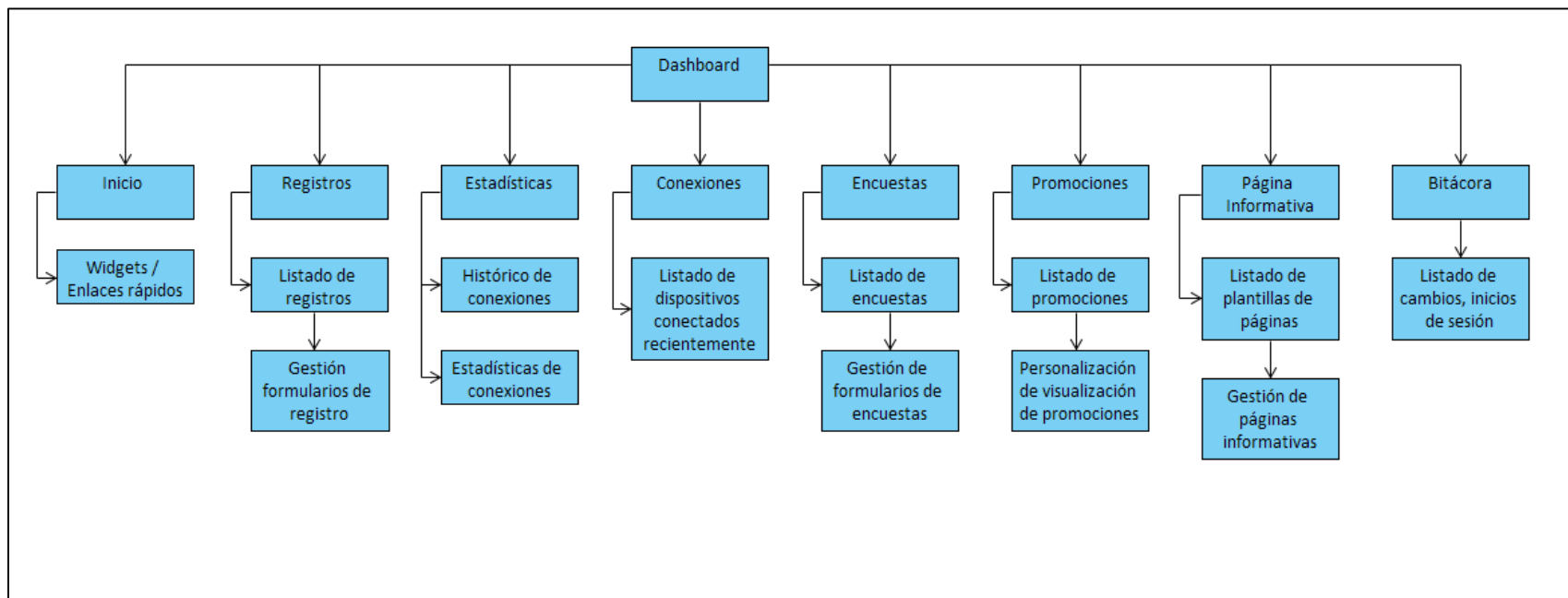


Imagen 23 Diagrama de navegación para usuario establecimiento cliente

- Diagrama de navegación para usuario general

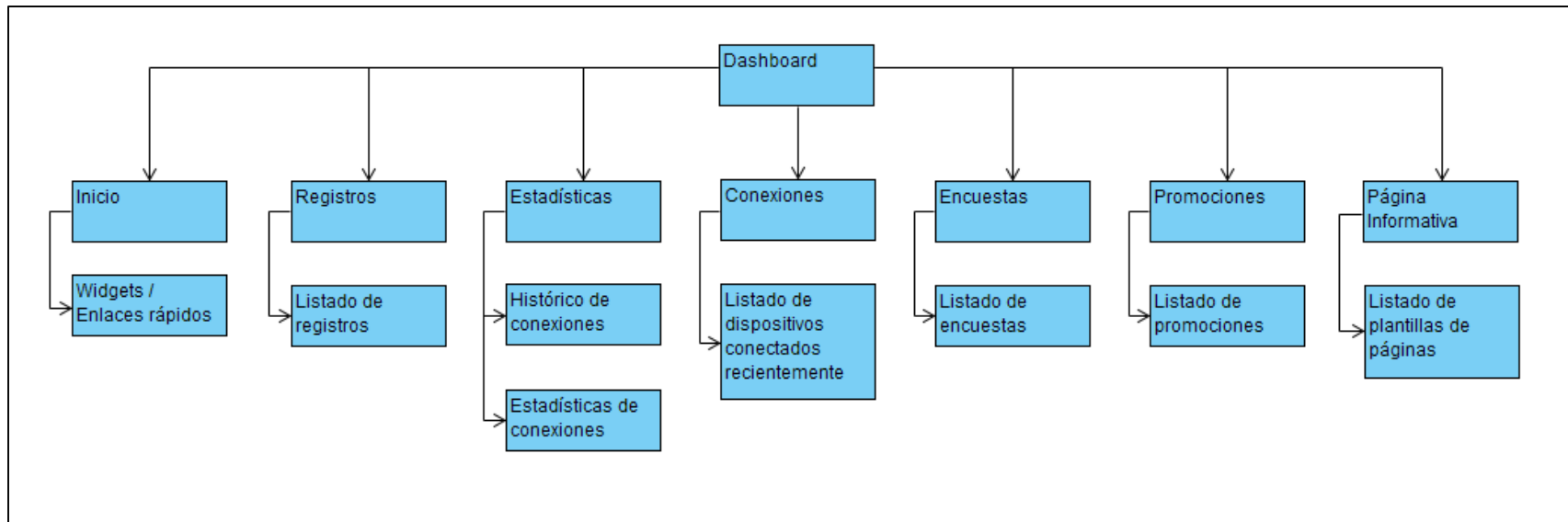


Imagen 24 Diagrama de navegación para usuario general

### 6.3.4 Diagrama lógico de la base de datos

Dado que para este proyecto se utilizó MongoDB, una base de datos No Relacional. Por tanto, se muestra un diagrama de las colecciones utilizadas para el desarrollo del proyecto.

La nomenclatura usada para el diagrama es la siguiente:

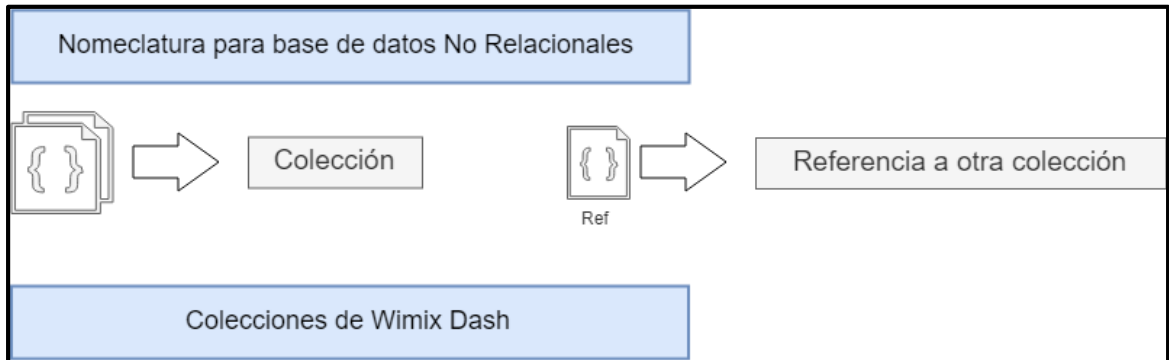


Imagen 25 Nomenclatura para la Base de Datos

A continuación, se muestran las colecciones utilizadas en ningún orden específico:

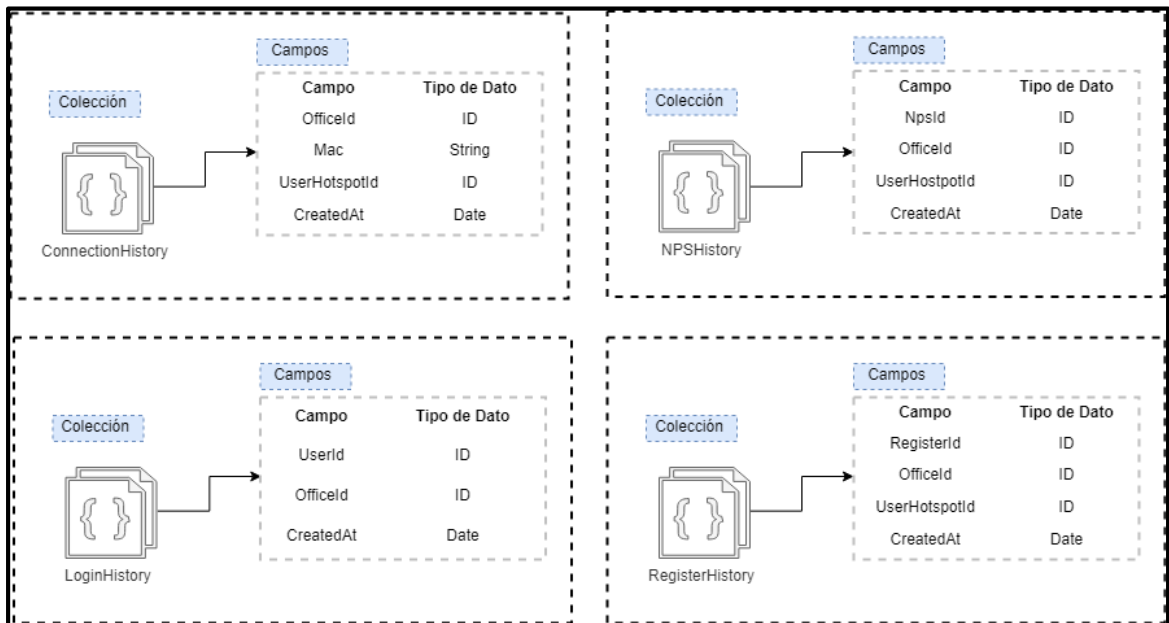


Imagen 26 Colecciones de la base de datos I

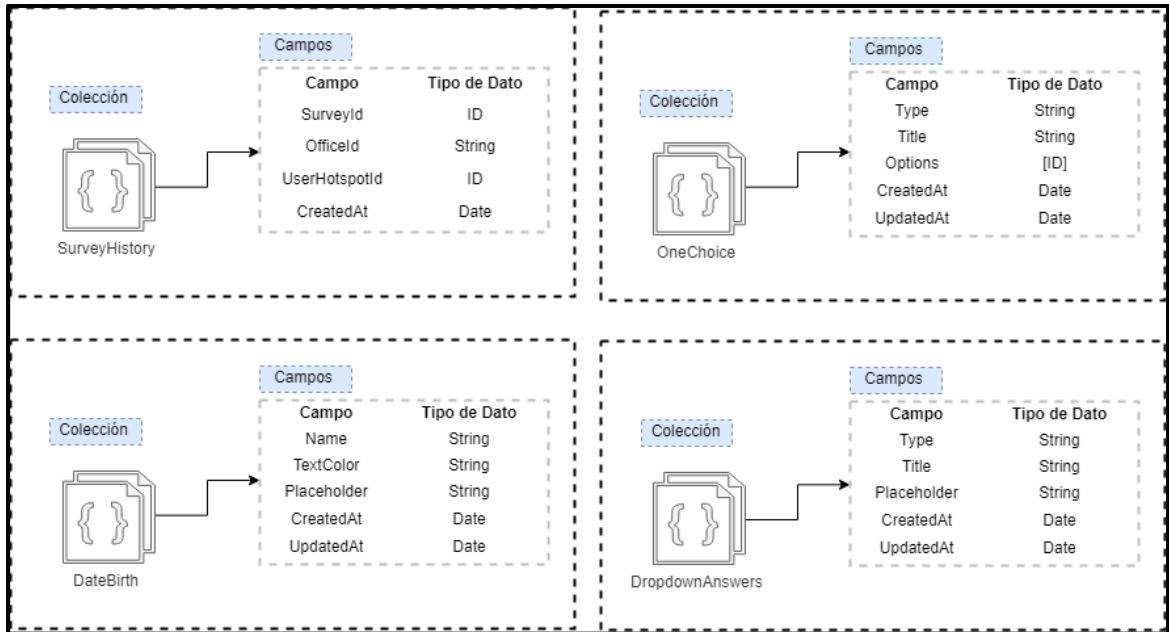


Imagen 27 Colecciones de la base de datos II

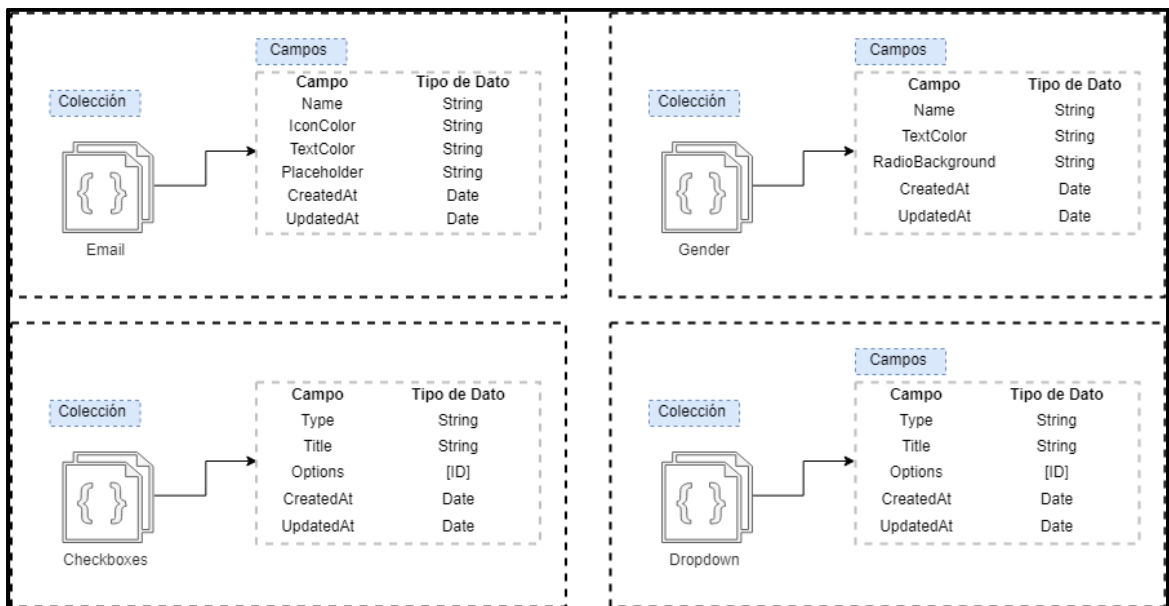


Imagen 28 Colecciones de la base de datos III

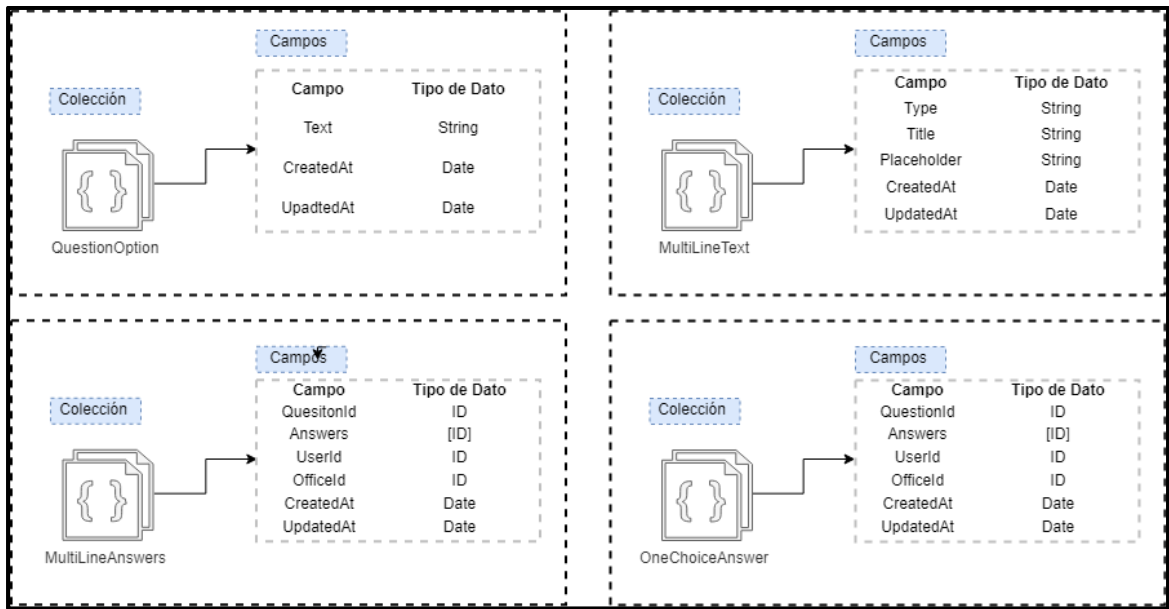


Imagen 29 Colecciones de la base de datos IV

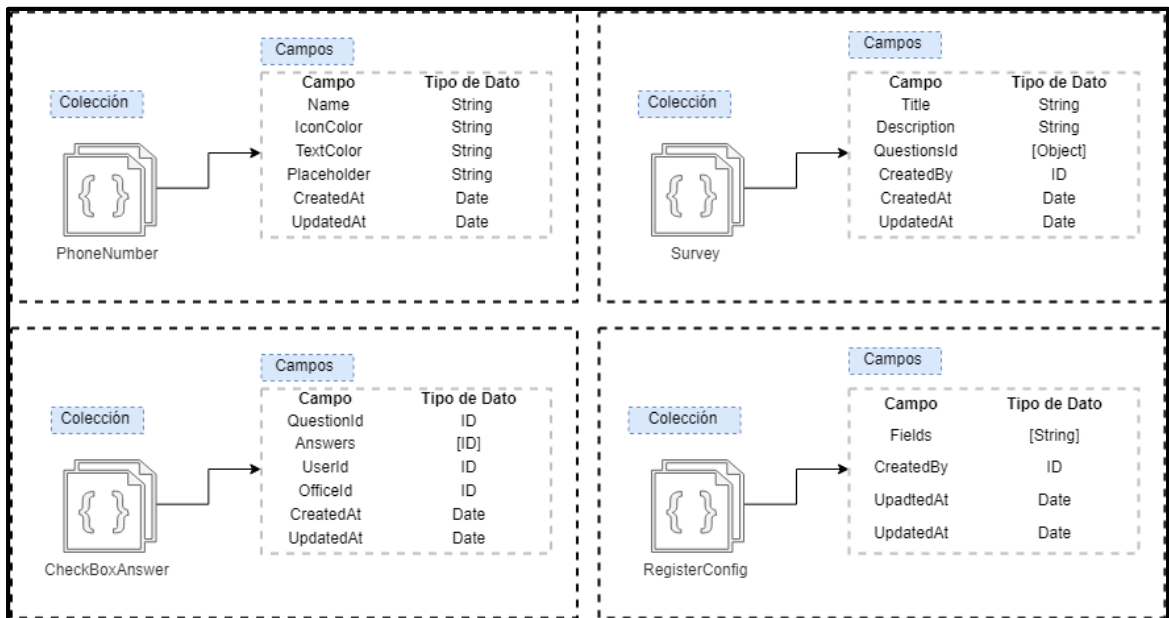


Imagen 30 Colecciones de la base de datos V

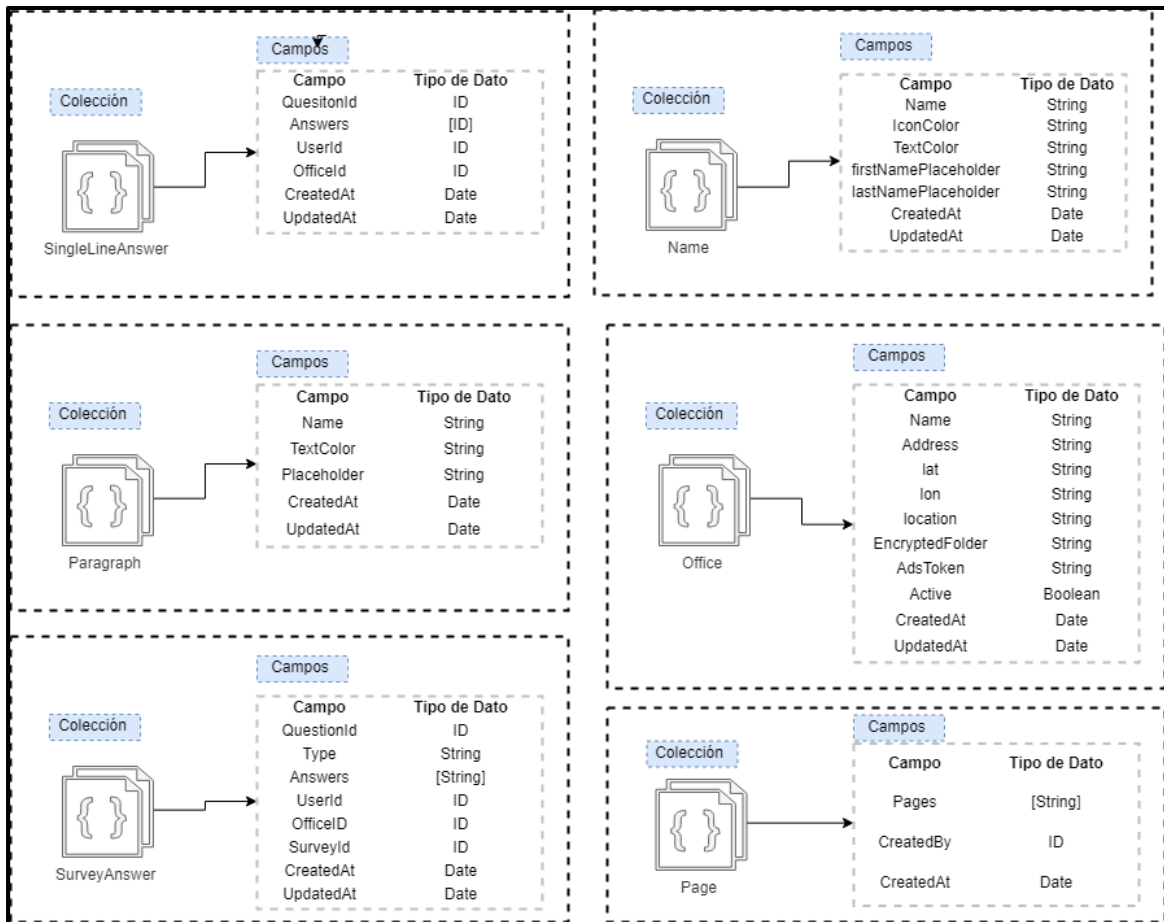


Imagen 31 Colecciones de la base de datos VI

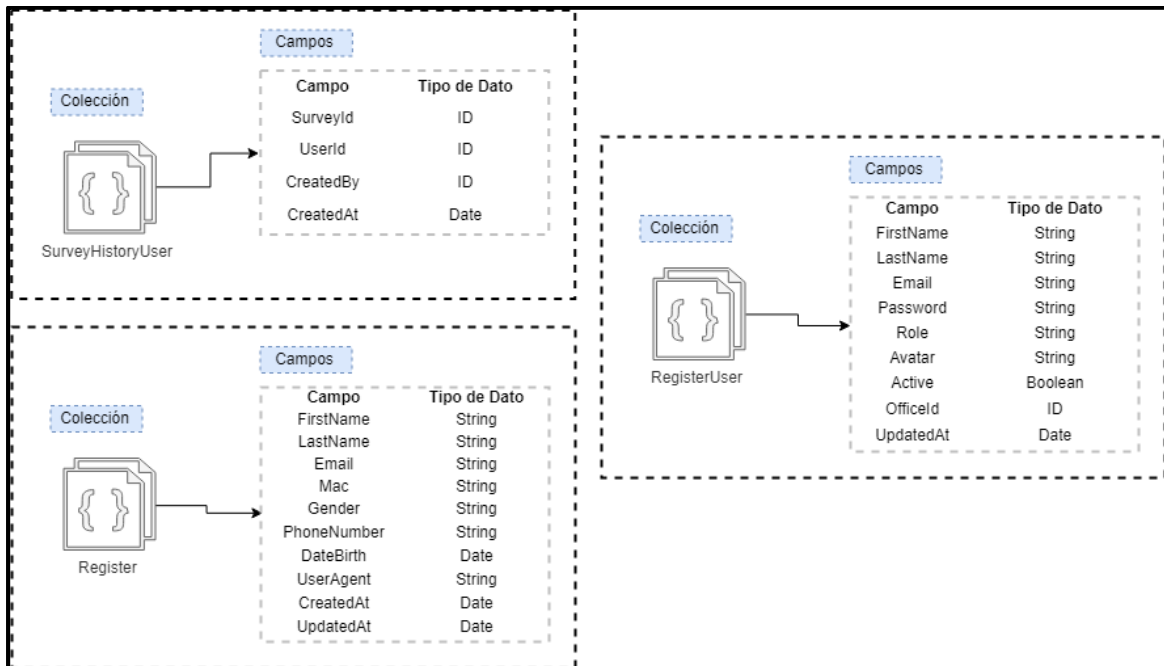
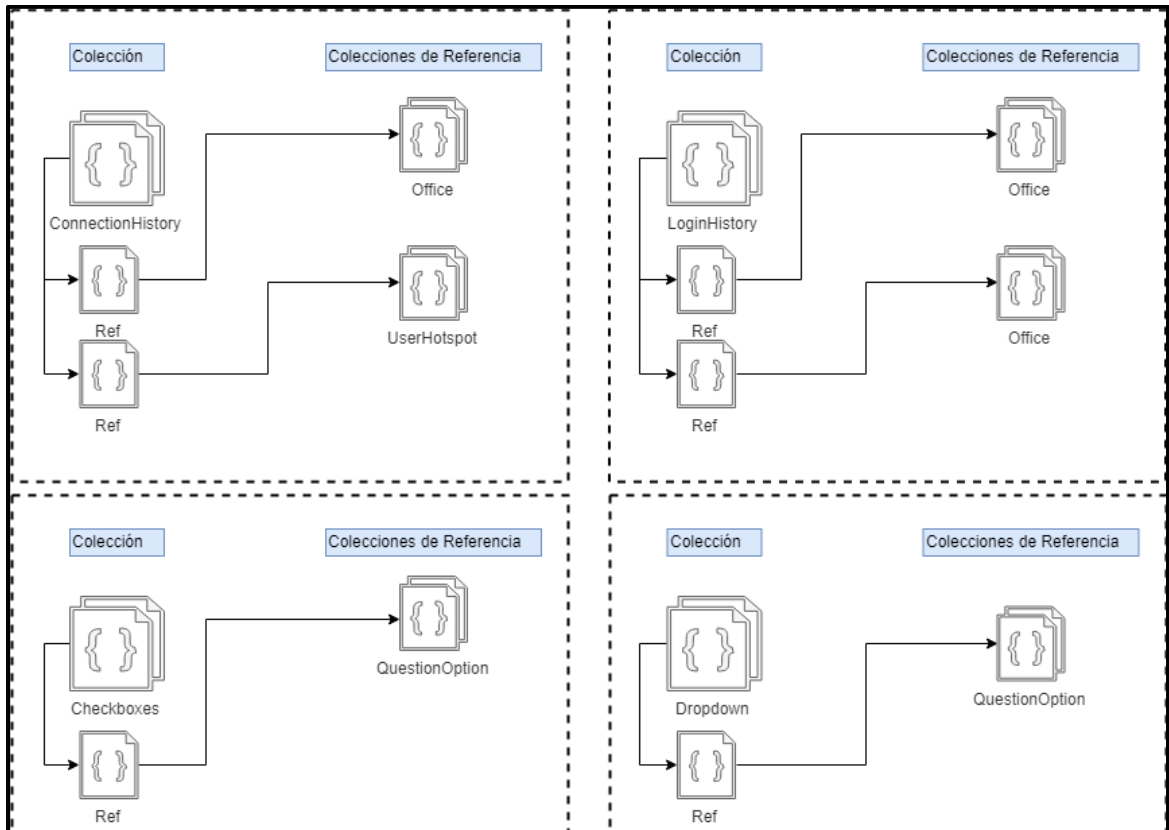


Imagen 32 Colecciones de la base de datos VII

En las bases de datos no relaciones, las colecciones pueden hacer referencias a otras colecciones. A continuación, se muestran las referencias hechas durante el desarrollo del sistema.



*Imagen 33 Relaciones por referencia entre colecciones I*



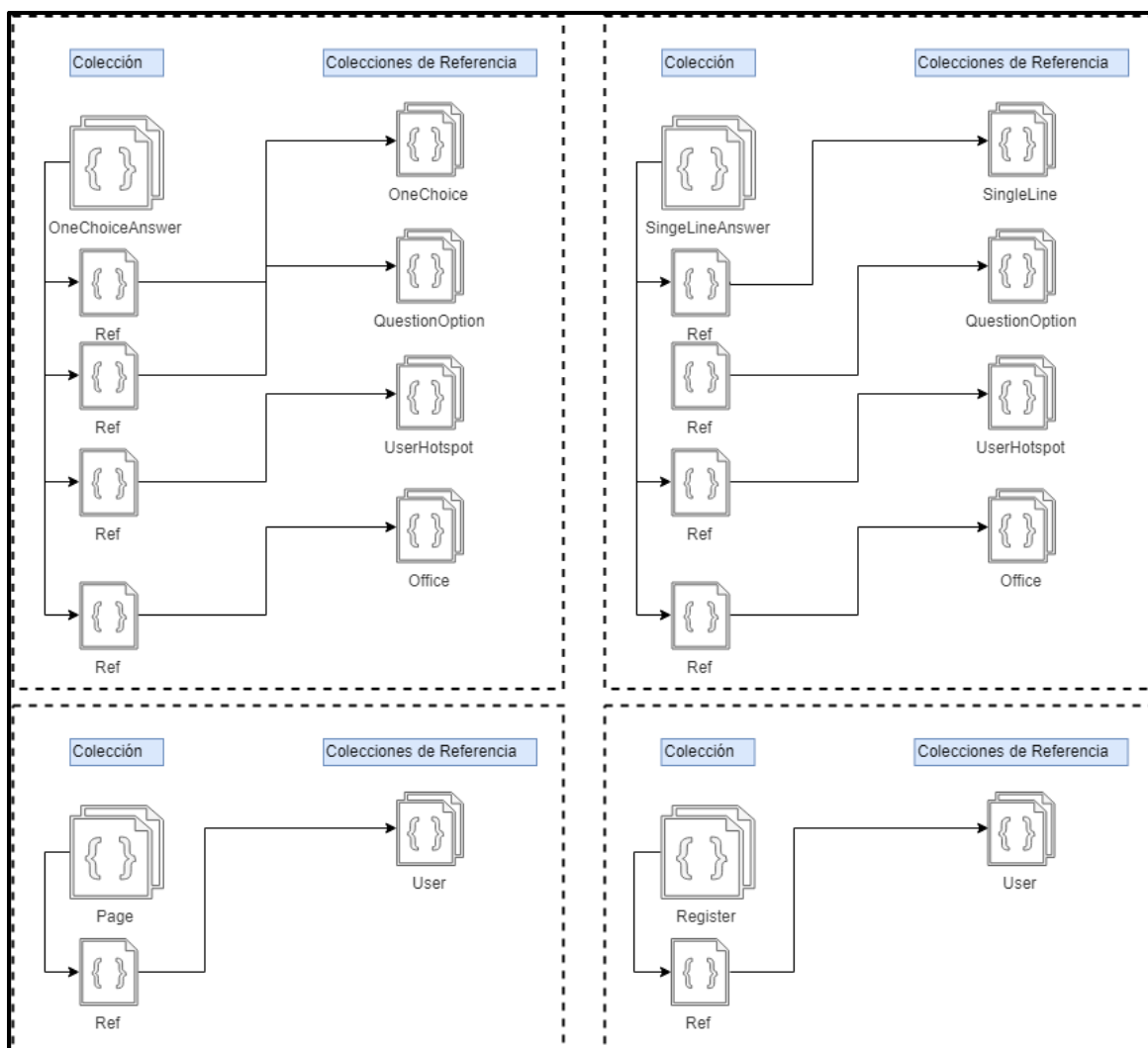


Imagen 34 Relaciones por referencia entre colecciones II

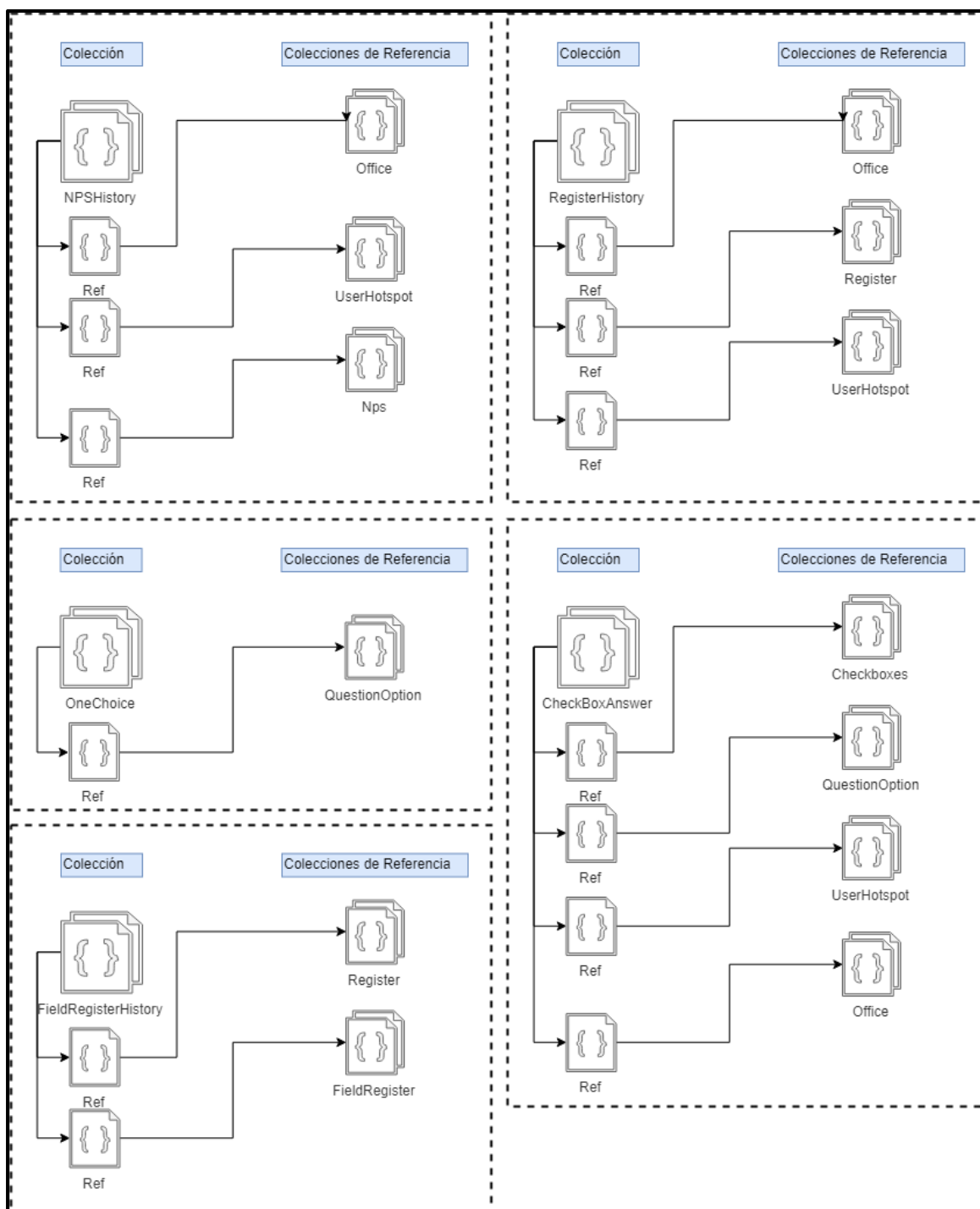


Imagen 35 Relaciones por referencias entre colecciones III

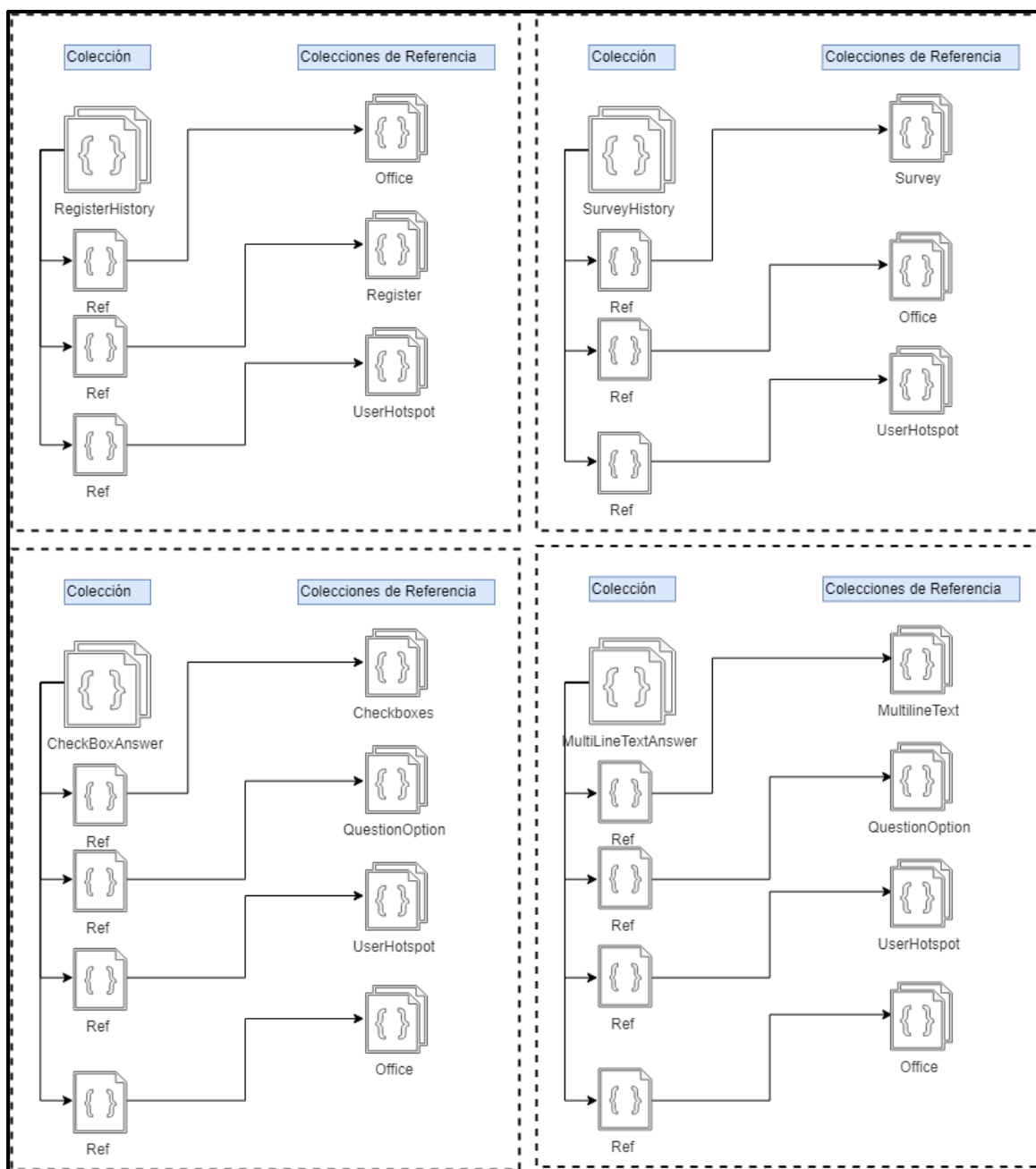


Imagen 36 Relaciones por referencia entre relaciones IV

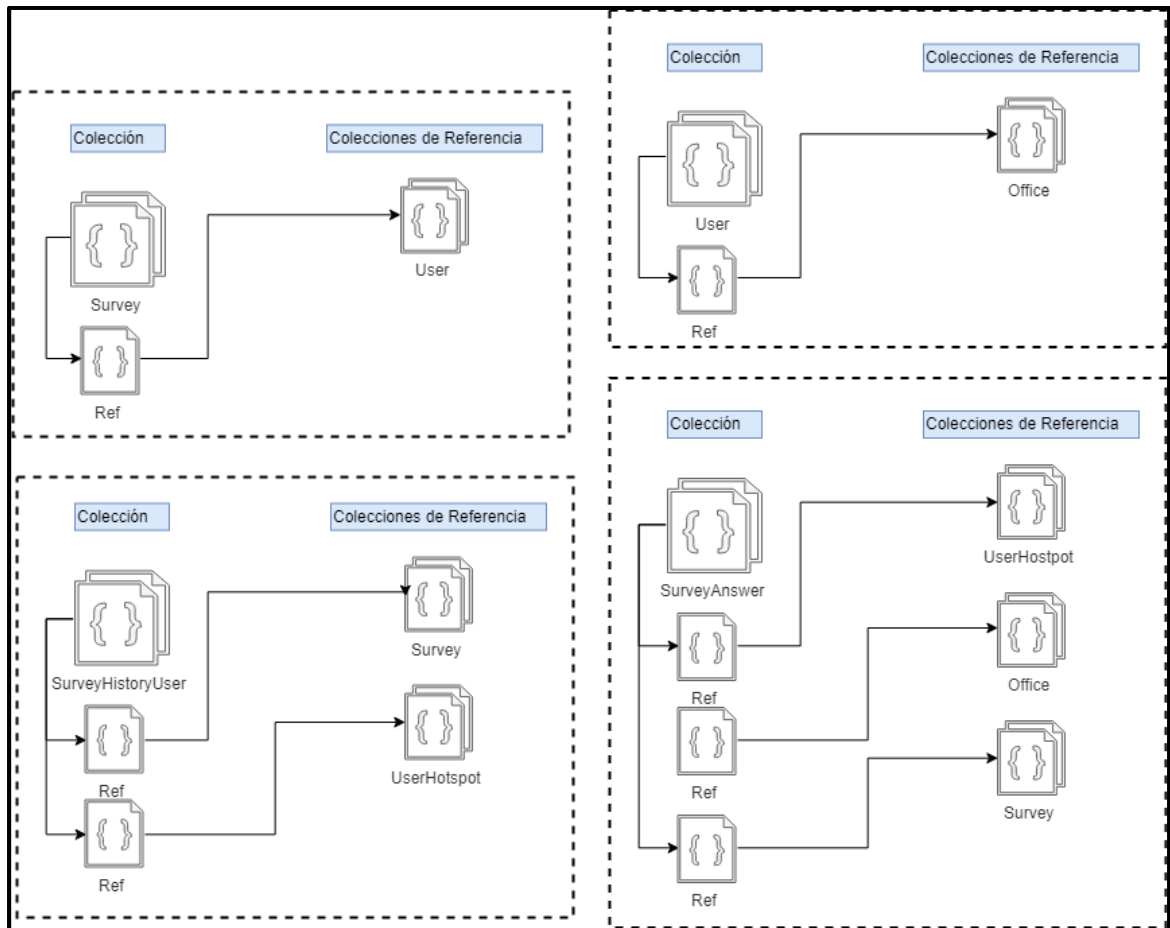


Imagen 37 Relaciones por referencia entre colecciones V

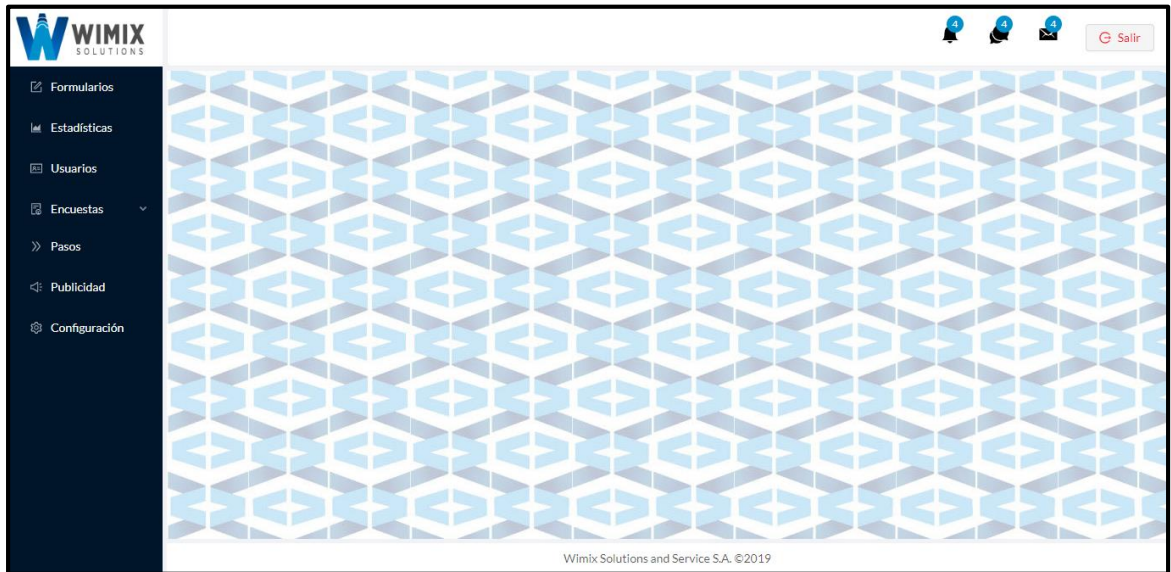
### 6.3.5 Diseño del Interfaz de Usuario

El diseño del sistema es simple y amigable, con el objetivo de crear un sistema de fácil uso para un usuario con los conocimientos mínimos de manejo de software, reduciendo así la posibilidad de errores y mejora de eficiencia. Se hizo uso de Ant Design, un lenguaje de diseño creado para facilitar la creación de aplicaciones web, también así seguir los estándares web de manera que un sistema, página o aplicación pueda ser vista en los distintos navegadores populares al momento.

A continuación, se describe cómo está constituido el sistema en base a las vistas:

- **Estructura principal:** En la imagen 24 se señalan los componentes básicos de la estructura principal del sistema, los cuales son componentes

cuyo contenido es dinámico; sin embargo, su ubicación es siempre la misma.



*Imagen 38 Diseño del sistema - Estructura principal*

- Logo: Logotipo de la empresa
- Notificaciones: Notificaciones de acceso rápido
- Menú lateral: Accesos a distintos módulos del sistema.
- Cuerpo: Contenedor, cambia el contenido dependiendo del módulo en el que se encuentre.
- Footer: Footer del sistema.

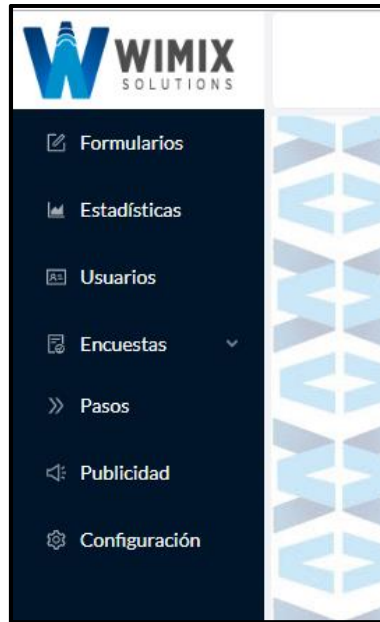


Imagen 39 Diseño del sistema - Menú lateral



Imagen 40 Diseño del sistema – Notificaciones

- **Módulos:** En la imagen 25 se muestra la estructura básica de un módulo de configuraciones. Manteniendo en la sección izquierda la vista previa de la configuración y en la sección derecha, las distintas opciones de configuración.



Imagen 41 Diseño del sistema - Módulos

- **Modales / Ventanas pop-ups:** Como parte de la mejora en experiencia de usuario, se utilizan modales o ventanas pop-ups, que facilitan las configuraciones, de manera que el usuario pueda cambiarlas sin necesidad de recargar la página.

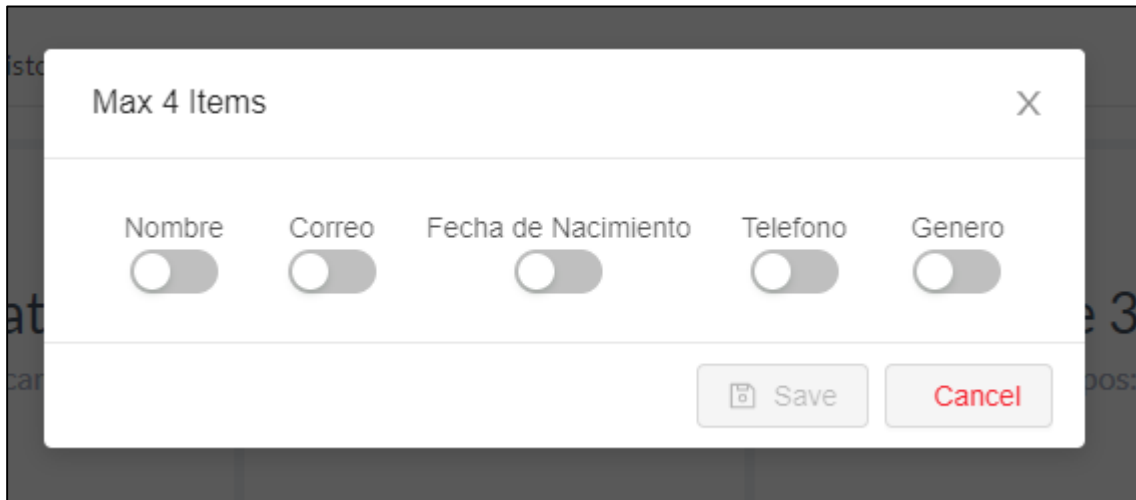


Imagen 42 Diseño del sistema - Modales

# Fase de Codificación



## 6.4. Fase de Codificación

### 6.4.1 Arquitectura de software

A inicio de proyecto se consideró como arquitectura de software el Modelo - Vista – Controlador. Sin embargo, con el cambio de tecnologías y herramientas a utilizar se consideró necesario cambiar también la arquitectura del software y su distribución a petición del cliente. Por tanto, se resolvió que la arquitectura o modelo más adecuado para la estructuración del software sería el Modelo Cliente – Servidor.

Es un modelo de diseño de software en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor, quien le da respuesta. Esta idea también se puede aplicar a programas que se ejecutan sobre una sola computadora, aunque es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras.

El proceso completo, desde que el usuario solicita una página, hasta que el cliente web (navegador) se la muestra con el formato apropiado, es el siguiente:

1. El usuario especifica en el cliente web la dirección de la página que desea consultar: el usuario escribe en el navegador la dirección (Universal Resource Locator (URL)) de la página que desea visitar.
2. El cliente establece una conexión con el servidor web.
3. El cliente solicita la página o el objeto deseado.
4. El servidor envía dicha página u objeto (o, si no existe, devuelve un código de error).
5. Si se trata de una página HTML, el cliente inicia sus labores de interpretación de los códigos HTML. Si el cliente web encuentra instrucciones que hacen referencia a otros objetos que se tienen que mostrar con la página (imágenes, sonidos, animaciones multimedia, etc.),

establece automáticamente comunicación con el servidor web para solicitar dichos objetos.

6. Se cierra la conexión entre el cliente y el servidor.
7. Se muestra la página al usuario.

Algunas de las ventajas ofrecidas por este modelo son:

- Centralización del control: los accesos, recursos y la integridad de los datos son controlados por el servidor de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema. Esta centralización también facilita la tarea de poner al día datos u otros recursos (mejor que en las redes P2P).
- Escalabilidad: se puede aumentar la capacidad de clientes y servidores por separado. Cualquier elemento puede ser aumentado (o mejorado) en cualquier momento, o se pueden añadir nuevos nodos a la red (clientes y/o servidores).
- Fácil mantenimiento: al estar distribuidas las funciones y responsabilidades entre varios ordenadores independientes, es posible reemplazar, reparar, actualizar, o incluso trasladar un servidor, mientras que sus clientes no se verán afectados por ese cambio (o se afectarán mínimamente). Esta independencia de los cambios también se conoce como encapsulación.
- Existen tecnologías, suficientemente desarrolladas, diseñadas para el paradigma de C/S que aseguran la seguridad en las transacciones, la amigabilidad de la interfaz, y la facilidad de empleo.

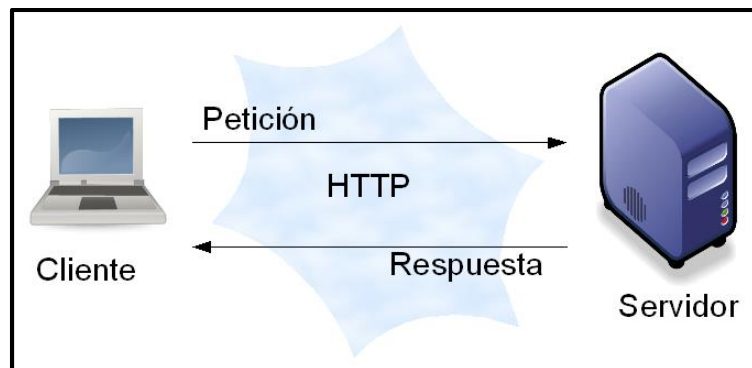


Imagen 43 Arquitectura del software

## 6.4.2 Estructura del sistema

A continuación, se presenta la estructura del código utilizado para el desarrollo del sistema. La carpeta principal del proyecto se divide en Server y Client.

En la tabla siguiente se muestra la carpeta principal del proyecto, donde se aprecian los archivos de cada una: configuraciones, esquemas, modelos, scripts, entre otros. Se explicará de manera general en qué consiste cada uno de ellos.

### 6.4.2.1 Server

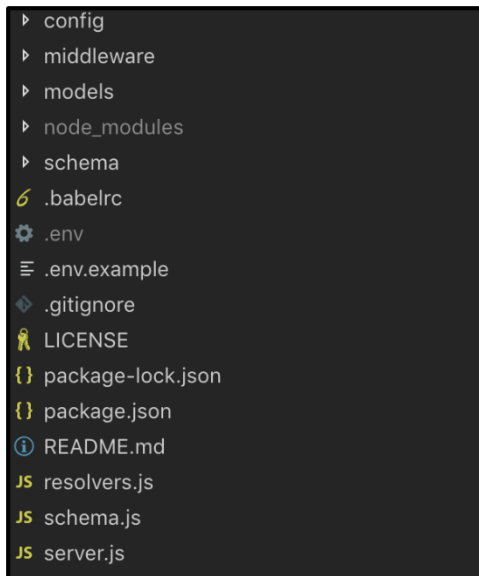


Imagen 44 Carpeta de archivos - Server

En la carpeta **server** tenemos los archivos necesarios para el correcto funcionamiento del back-end del proyecto. Descripción:

| Server |  |
|--------|--|
| Nombre | Explicación  |
| /      | <ul style="list-style-type: none"> <li>• <b>package.json</b>: Lista todas las dependencias necesarias para el desarrollo del back-end.</li> <li>• <b>server</b>: Archivo encargado de levantar el servidor del back-end.</li> <li>• <b>.gitignore</b>: Archivo encargado de evitar subir algunos archivos/carpetas al repositorio remoto y el servidor de producción.</li> </ul> |

| Server             |  |
|--------------------|--|
| <b>/config</b>     | Aquí se definen archivos de configuración para las variables de entorno, las cuales son necesarias en la petición de servicios externos o valores que por seguridad no deben estar a vista del repositorio, estas configuraciones toman en cuenta dos estados del proyecto: desarrollo y producción. |
| <b>/middleware</b> | Aquí se definen configuraciones necesarias para interceptar las peticiones que se hagan al back-end del proyecto, por ejemplo la autenticación del usuario.  |
| <b>/models</b>     | Aquí se definen los modelos correspondientes a la base de datos, cada colección, los documentos y subdocumentos que representan una entidad en la base de datos.   |
| <b>/schema</b>     | Aquí se definen las configuraciones necesarias para la construcción del back-end del proyecto, el esquema del tipo de información que se podrá hacer petición, así como también el código necesario que responda a la petición, creación, modificación y eliminación de datos                        |

Tabla 22 Carpeta de archivos - Server

#### 6.4.2.2 Client

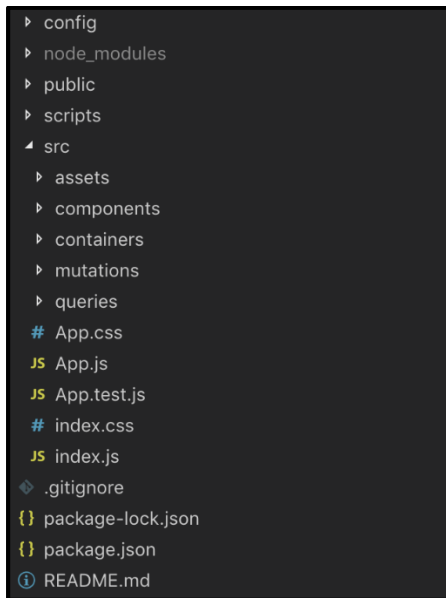


Imagen 45 Carpeta de archivos - Client

En la carpeta **client** tenemos los archivos necesarios para el correcto funcionamiento del backend del proyecto. Descripción:

| Client         |  |
|----------------|--|
| Nombre         | Explicación  |
| <b>/config</b> | Aquí se definen archivos de configuración para las variables de Esta carpeta contiene los archivos de configuración del cliente, |

| Client          |  |
|-----------------|--|
|                 | como son las pruebas, ambiente de desarrollo, transpiladores, organización de módulos.   |
| <b>/public</b>  | Esta carpeta contiene los archivos públicos al momento de lanzar a producción el front-end del proyecto.   |
| <b>/scripts</b> | Carpeta autogenerada para archivos que contienen ejecuciones de comandos de acuerdo a la interfaz de comando que se está utilizando.   |
| <b>/src</b>     | <p>Esta carpeta contiene subcarpetas, las cuales se detallan a continuación:</p> <ul style="list-style-type: none"> <li>• <b>Assets:</b> Archivos necesarios para recurso estáticos del proyecto como son imágenes, fuentes tipográficas, gif, estilos, etc.</li> <li>• <b>Components:</b> Estas carpetas contienen las vistas necesarias para cada módulo.</li> <li>• <b>Containers:</b> Contiene archivos que están relacionado con el dominio del proyecto, es decir, el modelo de negocio, interacciones, disparadores de eventos, datos modificados antes de ser mostrados en la vista.</li> <li>• <b>Mutations:</b> Contiene los archivos necesarios para enviar modificaciones a los datos del back-end.</li> <li>• <b>Queries:</b> Contiene los archivos encargados de llamar los datos del back-end.</li> </ul> |

Tabla 23 Carpeta de archivos – Client

Finalmente, se muestra en la siguiente imagen (Imagen 46 Estructura del sistema) las interacciones del sistema con los distintos lenguajes, servicios y dispositivos utilizados:

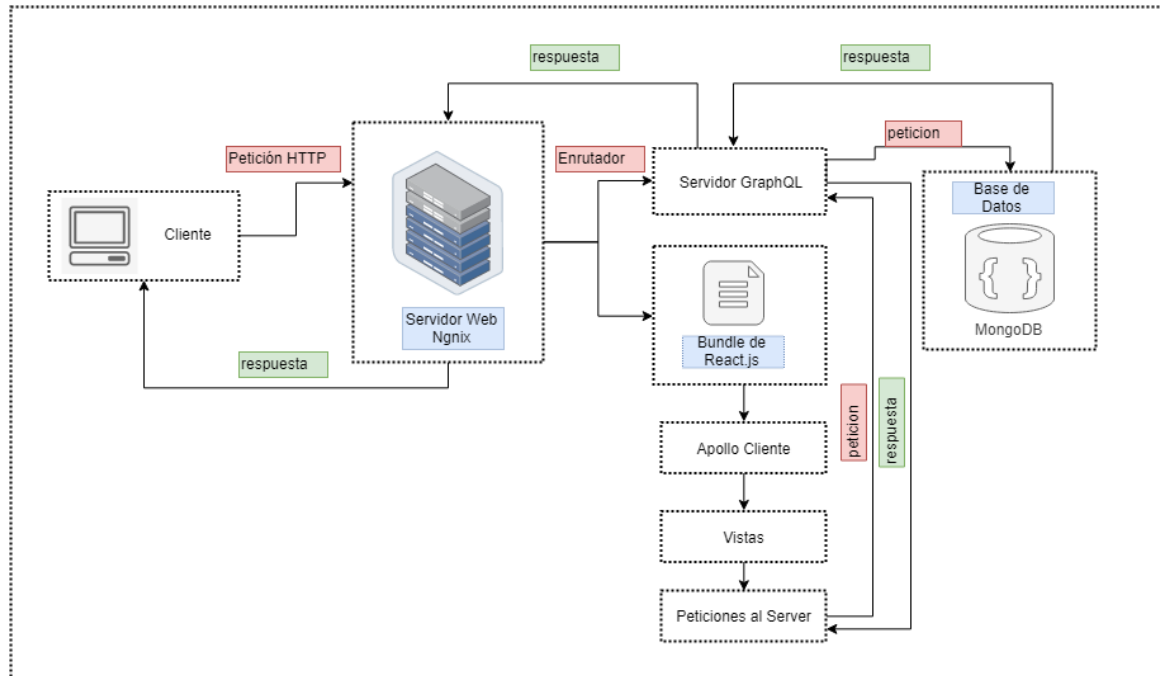


Imagen 46 Estructura del sistema

# Fase de pruebas

## 6.5. Fase de Prueba

El plan de pruebas se elabora con el fin de especificar qué elementos o componentes se van a probar para que el grupo de trabajo pueda realizar el proceso de Validación y Verificación de los requerimientos funcionales y no funcionales del sistema.

- **Estrategia de pruebas**



A través de las historias de usuario y el seguimiento de iteraciones antes realizados, se pretende retomar información directamente relacionada con las pruebas, para asegurar la calidad del sistema. Además, le permite al responsable de las pruebas saber los criterios se deben de tomar en cuenta al momento de probar cada elemento del sistema.

- **Alcance**

Teniendo en cuenta el plan de iteraciones y tareas, y en base a la especificación de las historias de usuario, el grupo de trabajo realizó las pruebas haciendo uso de las pruebas de funcionalidad, seguridad y tiempos de respuesta; permitiendo garantizar que el sistema cumpla con los requisitos mínimos de calidad acordados con Wimix, que, como resultado, no ocasionará errores inesperados cuando el sistema entre en producción.

- **Documentación de pruebas**

Para la documentación de las pruebas se procedió a utilizar la siguiente plantilla, basada en la plantilla proporcionada por (PMOinformatica, 2014):

| Pruebas                          |                                       |   |
|----------------------------------|---------------------------------------|---|
| Identificador: [ID de la prueba] | Historia de usuario: [ID HU asociado] | Eval.   |
| Nombre de caso de prueba:        |                                       |   |
| Fecha:                           |                                       |   |
| Descripción:                     |                                       |   |
| Condiciones de ejecución:        |                                       |   |
| Entradas:                        |                                       |   |



| Pruebas             |  |
|---------------------|--|
| Resultado esperado: |  |
| Resultado obtenido: |  |

Tabla 24 Plantilla de pruebas

### 6.5.1 Pruebas de funcionalidad

Las pruebas de funcionalidad aseguran el trabajo apropiado de los requisitos funcionales del sistema, incluyendo la navegación, entrada de datos, procesamiento y obtención de resultados.

- Iteración 1




| Pruebas   |                              |   |
|---|------------------------------|---|
| Identificador: P-001  | Historia de usuario: HU-01-A | Eval.   |
| Nombre de caso de prueba: Configurar y guardar formulario de registro. Con plantillas   |                              |  |
| Fecha: 21 Marzo 2019  |                              |   |
| Descripción: Esta prueba consiste en verificar si el sistema cumple con el requisito de configurar el formulario de registro que será utilizado por el establecimiento cliente. |                              |   |
| Condiciones de ejecución: Haber iniciado sesión con el usuario con permisos para realizar configuraciones   |                              |   |
| Entradas:<br>1. Elegir la opción “Formularios” del menú lateral.<br>2. Seleccionar una opción entre las distintas plantillas ofrecidas.<br>3. Dar clic en el botón Guardar.     |                              |   |
| Resultado esperado: Mensaje del sistema: “Guardado con éxito”   |                              |   |
| Resultado obtenido: El sistema mostró un mensaje de: “Guardado con éxito”   |                              |   |

Tabla 25 Prueba - Configurar y guardar formulario de registro

| Pruebas   |                              |   |
|---|------------------------------|---|
| Identificador: P-002  | Historia de usuario: HU-01-A | Eval.   |
| Nombre de caso de prueba: Configurar y guardar formulario de registro. Personalizado  |                              |  |
| Fecha: 21 Marzo 2019  |                              |   |
| Descripción: Esta prueba consiste en verificar si el sistema cumple con el requisito de configurar el formulario de registro que será utilizado por el establecimiento cliente. |                              |   |


| Pruebas   |  |
|---|--|
| <b>Condiciones de ejecución:</b> Haber iniciado sesión con el usuario con permisos para realizar configuraciones  |  |
| <b>Entradas:</b> <ol style="list-style-type: none"> <li>1. Elegir la opción “Formularios” del menú lateral.</li> <li>2. Seleccionar la opción de “Nueva Plantilla”</li> <li>3. En la ventana pop-up elegir como campos de prueba: <ul style="list-style-type: none"> <li>- Nombre</li> <li>- Correo</li> <li>- Género</li> </ul> </li> <li>4. Dar clic en el botón “Guardar” de la ventana.</li> <li>5. Una vez abierto el nuevo formulario, dar clic en el botón Guardar.</li> </ol> |  |
| <b>Resultado esperado:</b> Mensaje del sistema: “Guardado con éxito”  |  |
| <b>Resultado obtenido:</b> El sistema mostró un mensaje de: “Guardado con éxito”  |  |

Tabla 26 Configurar y guardar formulario de registro

| Pruebas   |                              |   |
|---|------------------------------|---|
| Identificador: P-003  | Historia de usuario: HU-02-A | Eval.   |
| Nombre de caso de prueba: Configurar y guardar encuestas NPS  |                              |  |
| Fecha: 21 Marzo 2019  |                              |   |
| Descripción: Esta prueba consiste en realizar las configuraciones necesarias para crear una encuesta NPS que será mostrada en el portal cautivo, y finalmente que guarde con éxito todos los cambios.   |                              |   |
| Condiciones de ejecución: Haber iniciado sesión con el usuario con permisos para realizar configuraciones   |                              |   |
| Entradas:<br>1. Seleccionar la opción “Encuestas > NPS” del menú principal<br>2. Seleccionar la opción “Áreas”<br>3. Dar clic en el botón “Nueva Etiqueta”.<br>4. Posteriormente escribir en el campo de texto que aparecerá: “Cajero”.<br>5. Repetir el paso número 3.<br>6. Escribir en el campo de texto: “Servicios Bancarios”<br>7. Repetir el paso número 3.<br>8. Escribir en el campo de texto “Transacciones”<br>9. Dar clic en el botón “Guardar” de la sección “Acciones”<br>10. Seleccionar la opción de “Puntuación”<br>11. Escribir en el campo de datos la pregunta deseada: “¿Con qué probabilidad recomendarías nuestros servicios a tus amigos y familiares?”. Dar clic en “Guardar”<br>12. Seleccionar la opción “Sugerencias”<br>13. Escribir en el campo de datos el texto deseado: “Por favor, argumenta el motivo de dicha evaluación”. Dar clic en el botón “Guardar” |                              |   |

| Pruebas   |  |
|---|--|
| 14. Dar clic en el botón “Guardar encuesta”   |  |
| <b>Resultado esperado:</b> El sistema muestra un mensaje diciendo: “Guardado con éxito” |  |
| <b>Resultado obtenido:</b> El sistema mostró un mensaje diciendo: “Guardado con éxito”  |  |

Tabla 27 Prueba - Configurar y guardar encuestas NPS

| Pruebas  |                              |   |
|--|------------------------------|---|
| Identificador: P-004   | Historia de usuario: HU-01-B | Eval.   |
| Nombre de caso de prueba: Configurar y guardar encuestas personalizadas  |                              |  |
| Fecha: 21 Marzo 2019   |                              |   |
| Descripción: Esta prueba consiste en realizar las configuraciones necesarias para crear una encuesta personalizada que será mostrada en el portal cautivo, y finalmente que guarde con éxito todos los cambios.  |                              |   |
| Condiciones de ejecución: Haber iniciado sesión con los roles y permisos correspondientes.   |                              |   |
| <b>Entradas:</b><br><ol style="list-style-type: none"><li>1. Seleccionar el menú “Encuestas &gt; Personalizadas”</li><li>2. Ingresar en el primer campo de texto un título para la encuesta: “Sistema de tickets”</li><li>3. Ingresar en el siguiente campo de texto una descripción para la encuesta: “Encuesta para evaluar satisfacción del cliente ante el nuevo sistema de tickets”</li><li>4. Dar clic en el botón “Siguiente”</li><li>5. Seleccionar distintos tipos de campos:<ul style="list-style-type: none"><li>- Una sola elección</li><li>- Múltiple línea de texto</li><li>- Selección con dropdown</li></ul></li><li>6. Dar clic en el botón “Siguiente”</li><li>7. Seleccionar la opción “Una sola elección”. Escribir un título: “¿Ha utilizado nuestro sistema de tickets?”. Ingresar 2 opciones: “Sí” y “No”. Dar clic en el botón “Guardar”</li><li>8. Seleccionar la opción “Múltiple líneas de texto”. Escribir un título: “Describa su experiencia”. Ingresar en el siguiente cuadro: “Escriba su opinión...” Dar clic en el botón “Guardar”</li><li>8. Seleccionar la opción “Selección con Dropdown”. Escribir un título: “Selecciona un nivel de satisfacción (5 es excelente)”. Ingresar los números de 1 al 5 para las opciones del dropdown. Dar clic en el botón “Guardar”</li><li>9. Dar clic en el botón “Siguiente”. Pre visualizar la encuesta antes de guardar.</li><li>10. Dar clic en el botón “Guardar”</li></ol> |                              |   |
| Resultado esperado: El sistema mostrará un mensaje: “Guardado con éxito”   |                              |   |

| Pruebas   |  |
|---|--|
| <b>Resultado obtenido:</b> El sistema no ha mostrado el mensaje de “Guardado con éxito” |  |

Tabla 28 Configurar y guardar encuestas personalizadas

- Iteración 2


| Pruebas   |                              |   |
|---|------------------------------|---|
| Identificador: P-005  | Historia de usuario: HU-01-B | Eval.   |
| Nombre de caso de prueba: Verificar las conexiones recientes  |                              |  |
| Fecha: 21 Marzo 2019  |                              |   |
| Descripción: La siguiente prueba consiste en verificar las conexiones recientes (dentro de un rango mínimo de 20min)                                  |                              |   |
| Condiciones de ejecución: Haber iniciado sesión con los roles y permisos correspondientes.<br>Tener un dispositivo móvil de prueba conectado a la red |                              |   |
| Entradas:<br>1. Seleccionar el menú “Usuarios”<br>2. Verificar si el nuevo dispositivo aparece en la lista.   |                              |   |
| Resultado esperado: El sistema mostrará en pantalla los distintos dispositivos conectados.  |                              |   |
| Resultado obtenido: El sistema ha mostrado los dispositivos conectados al momento.  |                              |   |

Tabla 29 Prueba - Verificar las conexiones recientes




| Pruebas  |                              |   |
|--|------------------------------|---|
| Identificador: P-006   | Historia de usuario: HU-02-B | Eval.   |
| Nombre de caso de prueba: Guardar configuración de visualización de promociones  |                              |  |
| Fecha: 4 abril 2019  |                              |   |
| Descripción: En esta prueba se comprobará la funcionalidad del guardado para la visualización de las promociones en el portal cautivo. |                              |   |
| Condiciones de ejecución: Haber iniciado sesión con los roles y permisos correspondientes.   |                              |   |
| Entradas:<br>1. Seleccionar del menú lateral la opción “Promociones”<br>2. Seleccionar la opción deseada: “Horizontal”                 |                              |   |
| Resultado esperado: El sistema mostrará un mensaje de “Modo horizontal”  |                              |   |
| Resultado obtenido: El sistema ha mostrado un mensaje: “Modo Horizontal”. Verificado en el portal cautivo.                             |                              |   |

Tabla 30 Prueba - Guardar configuración de visualización de promociones


| Pruebas   |                              |   |
|---|------------------------------|---|
| Identificador: P-007  | Historia de usuario: HU-03-B | Eval.   |
| Nombre de caso de prueba: Exportar archivo de usuarios registrados  |                              |  |
| Fecha: 4 de abril 2019  |                              |   |
| Descripción: La siguiente prueba tiene como finalidad demostrar la funcionalidad exportar. Los registros de los usuarios registrados en un tiempo determinado, con o sin filtros, podrán ser exportados a un archivo Excel donde el usuario podrá hacer uso de la información según convenga. |                              |   |
| Condiciones de ejecución: Haber iniciado sesión con los roles y permisos correspondientes.  |                              |   |
| Entradas:<br>1. Visualizar la lista de usuarios registrados<br>2. Utilizar filtros y búsqueda<br>3. Exportar archivo dando clic en el botón “Descargar”   |                              |   |
| Resultado esperado: El sistema guardará en el dispositivo el archivo con los datos filtrados y buscados.  |                              |   |
| Resultado obtenido: El sistema ha arrojado un archivo que contiene la lista de todos los usuarios registrados hasta el momento, en dependencia de los filtros utilizados.   |                              |   |

*Tabla 31 Prueba - Exportar archivo de usuarios registrados*

- **Iteración 3**

| Pruebas   |                              |   |
|---|------------------------------|---|
| Identificador: P-008  | Historia de usuario: HU-04-C | Eval.   |
| Nombre de caso de prueba: Configuración de orden de las secciones pertenecientes al portal cautivo                                      |                              |  |
| Fecha: 4 abril 2019   |                              |   |
| Descripción: La prueba se realiza para verificar la funcionalidad de guardado del orden de las secciones del portal cautivo             |                              |   |
| Condiciones de ejecución: Haber iniciado sesión con los roles y permisos correspondientes   |                              |   |
| Entradas:<br>1. Seleccionar y arrastrar las secciones en orden para que el portal cautivo muestre al usuario<br>2. Guardar              |                              |   |
| Resultado esperado: En el dispositivo conectado, se mostrarán las secciones en el orden exacto en que se guardó en las configuraciones. |                              |   |
| Resultado obtenido:   |                              |   |

*Tabla 32 Prueba - Configuración de orden de las secciones pertenecientes al portal cautivo*

| Pruebas  |                              |   |
|--|------------------------------|---|
| Identificador: P-009   | Historia de usuario: HU-01-B | Eval.   |
| Nombre de caso de prueba: Segunda Prueba: Configurar y guardar encuestas personalizadas  |                              |  |
| Fecha: 04 abril 2019   |                              |   |
| Descripción: Esta prueba consiste en realizar las configuraciones necesarias para crear una encuesta personalizada que será mostrada en el portal cautivo, y finalmente que guarde con éxito todos los cambios. La prueba realizada anteriormente no fue exitosa. Se realizaron los cambios correspondientes.  |                              |   |
| Condiciones de ejecución: Haber iniciado sesión con los roles y permisos correspondientes.   |                              |   |
| <b>Entradas:</b><br><ol style="list-style-type: none"><li>1. Seleccionar el menú “Encuestas &gt; Personalizadas”</li><li>2. Ingresar en el primer campo de texto un título para la encuesta: “Sistema de tickets”</li><li>3. Ingresar en el siguiente campo de texto una descripción para la encuesta: “Encuesta para evaluar satisfacción del cliente ante el nuevo sistema de tickets”</li><li>4. Dar clic en el botón “Siguiente”</li><li>5. Seleccionar distintos tipos de campos:<ul style="list-style-type: none"><li>- Una sola elección</li><li>- Múltiple línea de texto</li><li>- Selección con dropdown</li></ul></li><li>6. Dar clic en el botón “Siguiente”</li><li>7. Seleccionar la opción “Una sola elección”. Escribir un título: “¿Ha utilizado nuestro sistema de tickets?”. Ingresar 2 opciones: “Sí” y “No”. Dar clic en el botón “Guardar”</li><li>8. Seleccionar la opción “Múltiple líneas de texto”. Escribir un título: “Describa su experiencia”. Ingresar en el siguiente cuadro: “Escriba su opinión...” Dar clic en el botón “Guardar”</li><li>8. Seleccionar la opción “Selección con Dropdown”. Escribir un título: “Selecciona un nivel de satisfacción (5 es excelente)”. Ingresar los números de 1 al 5 para las opciones del dropdown. Dar clic en el botón “Guardar”</li><li>9. Dar clic en el botón “Siguiente”. Pre visualizar la encuesta antes de guardar.</li><li>10. Dar clic en el botón “Guardar”</li></ol> |                              |   |
| Resultado esperado: El sistema mostrará un mensaje: “Guardado con éxito”   |                              |   |
| Resultado obtenido: El sistema no ha mostrado el mensaje de “Guardado con éxito”   |                              |   |

## 6.5.2 Pruebas de seguridad

Las pruebas de seguridad consisten en probar los atributos o características de seguridad del sistema: si es vulnerable, si existe control de acceso, si dichos accesos pueden ser vulnerados.


| Pruebas  |                              |   |
|--|------------------------------|---|
| Identificador: P-010   | Historia de usuario: HU-04-A | Eval.   |
| Nombre de caso de prueba: Ingresar al sistema - Autenticación  |                              |  |
| Fecha: 4 de abril 2019   |                              |   |
| Descripción: Esta prueba consiste en verificar si el sistema cuenta con la seguridad básica para no permitir usuarios no registrados y que cada usuario cuente con los permisos debidos de acuerdo al rol establecido. |                              |   |
| Condiciones de ejecución: Abrir el sistema en la sección de inicio de sesión.  |                              |   |
| Entradas:<br>1. El usuario ingresará su correo y contraseña.<br>- Usuario: ramirez.salgado@wimixsolutions.com<br>- Contraseña: kelvin_ram@2019<br>2. El usuario hace clic en el botón Ingresar                         |                              |   |
| Resultado esperado: El sistema muestra un mensaje de: “Error al intentar ingresar al Sistema. Usuario y/o Contraseña Incorrecta”   |                              |   |
| Resultado obtenido: La persona no ha ingresado al sistema.   |                              |   |

Tabla 33 Prueba - Ingresar al sistema – Autenticación


| Pruebas  |                              |   |
|--|------------------------------|---|
| Identificador: P-011   | Historia de usuario: HU-04-A | Eval.   |
| Nombre de caso de prueba: Ingresar al sistema - Autenticación  |                              |  |
| Fecha: 4 de abril 2019   |                              |   |
| Descripción: Esta prueba consiste en verificar si el sistema cuenta con la seguridad básica para no permitir usuarios no registrados y que cada usuario cuente con los permisos debidos de acuerdo al rol establecido.                       |                              |   |
| Condiciones de ejecución: Abrir el sistema en la sección de inicio de sesión.  |                              |   |
| Entradas:<br>1. El usuario ingresará su correo y contraseña. <ul style="list-style-type: none"><li>- Usuario: kelvin.ramirez@wimixsolutions.com</li><li>- Contraseña: kelvin_ram@2019</li></ul> 2. El usuario hace clic en el botón Ingresar |                              |   |
| Resultado esperado: El sistema lleva al usuario a la pantalla de inicio del sistema.   |                              |   |
| Resultado obtenido: El usuario ha ingresado al sistema.  |                              |   |

Tabla 34 Prueba - Ingresar al Sistema - Autenticación II

# Fase de producción



## 6.6. Fase de producción

| ACTA DE ENTREGA   |                        |              |                  |        |
|---|------------------------|--------------|------------------|--------|
| Control de versiones  |                        |              |                  |        |
| Versión   | Revisada por           | Aprobada por | Fecha de entrega | Motivo |
| 1   |                        |              |                  |        |
| <b>Información General</b>  |                        |              |                  |        |
| <b>Nombre del proyecto</b>  |                        |              |                  |        |
| Sistema de gestión de contenido para portales cautivos y optimización del panel de administración de conexiones WiFi  |                        |              |                  |        |
| <b>Nombre del cliente</b>   |                        |              |                  |        |
| Wimix Solutions and Services  |                        |              |                  |        |
| <b>Equipo del proyecto</b>  |                        |              |                  |        |
|   |                        |              |                  |        |
| <b>Declaración de la Aceptación Formal</b>  |                        |              |                  |        |
| <p>Por medio de la presente acta, se deja constancia de la aceptación por parte de la Gerencia de Wimix Solutions and Services, del proyecto “Sistema de gestión de contenido para portales cautivos y optimización del panel de administración de conexiones Wifi”.</p> <p>En este punto se da por concluido el proyecto, por lo que habiendo constatado la Gerencia de Wimix Solutions and Services y el equipo de desarrollo, la finalización, entrega y aceptación del sistema se certifica el cierre de proyecto, el cual culmina de manera exitosa.</p> |                        |              |                  |        |
| <b>Aceptado por</b>   |                        |              |                  |        |
| <b>Nombre</b>   | <b>Cargo</b>           | <b>Firma</b> |                  |        |
| <b>Christian Burgos</b>   | Gerente de Operaciones |              |                  |        |
| <b>Entregado por</b>  |                        |              |                  |        |
| <b>Nombre</b>   |                        | <b>Firma</b> |                  |        |
| <b>Natalia Flores P.</b>  |                        |              |                  |        |
| <b>Billi Ramírez Obregón</b>  |                        |              |                  |        |

Tabla 35 Acta de entrega

# Capítulo IV:

## Conclusiones y Recomendaciones

## **VII. CONCLUSIONES Y RECOMENDACIONES**

Se logró desarrollar el sistema de gestión de contenido y la optimización del panel actual de Wimix Dash, utilizando la metodología XP y la Integración Continua / Despliegue Continuo, lo cual simplificó el proceso del proyecto, tanto en esfuerzo como en costos para la empresa Wimix Solutions and Services.

Las tecnologías y arquitectura utilizadas en este servicio representaron un alto valor para los usuarios de Wimix, dado que podían ingresar de forma recurrente y realizar los cambios necesarios en el momento preciso, sin generar ningún conflicto de acceso a los datos.

La seguridad de los datos de usuarios es muy importante en este tipo de sistemas dado que la información manejada en el sistema es medianamente sensible e implica riesgos a ataques informáticos. Por lo que la implementación de autenticación de usuario mediante un nombre de usuario y contraseña, fue necesaria

El sistema cumple con los objetivos propuestos, de los cuales se destacan particularmente los siguientes:

- Optimizar el diseño lógico del sistema por medio de componentes reutilizables para definir patrones de solución a nuevos requerimientos del usuario.
  - Gracias a la debida exploración de datos y lógica del negocio implantada al inicio de este proyecto monográfico, se pudo realizar un estudio de los componentes reutilizables en el sistema. Creando así un servicio ligero y más sencillo de usar.
- Codificar los componentes del sistema bajo plataformas de desarrollo web.
  - Los componentes reutilizables fueron creados gracias a las distintas herramientas utilizadas a lo largo de este proyecto. Dando pauta así de crear un sistema donde Wimix pueda habilitar o deshabilitar un módulo dependiendo de las necesidades de un cliente específico.
- Realizar pruebas del sistema para verificar su correcto funcionamiento.

- Las pruebas recurrentes dieron pauta a la retroalimentación con la empresa Wimix. Mediante ellas y la utilización de Integración Continua / Despliegue Continuo se logró realizar una por una las pruebas de usuario escritas al inicio del proyecto.
- Implantar el CMS en la empresa Wimix para sea accedido por sus clientes.

### **Recomendaciones:**

- Se recomienda la creación de un panel para la gestión de correos electrónicos.
- Se recomienda la mejora en la arquitectura de la aplicación, mediante la orquestación de servicios, siguiendo el patrón que utiliza Docker.
- Se recomienda optar por una mejor instancia de base de datos, con mejores recursos, ya que con la que se cuenta es de versión gratis, por lo cual los recursos son limitados.
- Se recomienda crear un equipo de desarrollo para el mantenimiento del sistema, siguiendo las fases de la metodología XP o Scrum, manteniendo un ambiente agile en el ciclo de vida del proyecto.
- Se recomienda integrar prácticas de Integración Continua y Despliegue Continuo, permitiendo entrega constante de nuevas funcionalidades, que aporten más valor agregado al sistema.
- Se recomienda la capacitación del sistema a clientes de la empresa Wimix a medida que el sistema este sujeto a cambios.
- Se propuso la compra de un certificado SSL para cubrir subdominios. Sin embargo, éste no ha sido comprado por la empresa Wimix. Es de mucha importancia para el lanzamiento oficial de la aplicación ya que permite agregar una capa mayor de seguridad al sistema.

# Capítulo V:

## Referencias

## VIII. REFERENCIAS

- 1&1 *Digital Guide*. (2017 de abril de 2017). Obtenido de <https://www.1and1.es/digitalguide/hosting/cms/cms-en-comparativa-los-gestores-de-contenido-mas-usados/#c53394>
- Amazon Web Services, Inc. (2019). *¿Qué es NoSQL?* Obtenido de Amazon Web Services: <https://aws.amazon.com/es/nosql/>
- Ant Design. (2019). *Ant Design Introduction*. Obtenido de Ant Design: <http://ant.design/docs/spec/introduce>
- Aplyca. (22 de Mayo de 2018). *GraphQL - Evolución del diseño de API's*. Obtenido de Aplyca Tecnología: <https://www.aplyca.com/es/blog/graphql>
- Berberá, J. (2014). *Clasificación y características de los CMS o Gestores de contenidos*. Obtenido de Horizonweb.es: <http://www.xn--diseowebmurcia1-1qb.es/clasificacion-caracteristicas-cms-gestores-de-contenidos/>
- Bhuvaneswari, N., & Sujatha, S. (2011). *Integrating SOA and Web Services*. River Publishers.
- Blog Bitix. (10 de Julio de 2015). *Crear y usar un repositorio en Docker Hub*. Obtenido de Blog Bitix: <https://picodotdev.github.io/blog-bitix/2015/07/crear-y-usar-un-repositorio-en-docker-hub/>
- Casado Martínez, C., & Córcoles, C. (10 de Mayo de 2008). *Crear y mantener una página web*. Obtenido de mosaic: <http://mosaic.uoc.edu/2008/05/10/crear-y-mantener-una-pagina-web/>
- Castro, R. (septiembre de 2005). *RedIRIS*. Obtenido de RedIRIS: <http://www.rediris.es/difusion/publicaciones/boletin/73/ENFOQUE1.pdf>
- Comunidad Developer Mozilla. (18 de Marzo de 2019). *Control de acceso HTTP (CORS)*. Obtenido de Developers Mozilla : [https://developer.mozilla.org/es/docs/Web/HTTP/Access\\_control\\_CORS](https://developer.mozilla.org/es/docs/Web/HTTP/Access_control_CORS)

- Culturación. (s.f.). *Frontend y backend: ¿Qué son?* Obtenido de Culturación: <http://culturacion.com/frontend-y-backend-%C2%BFque-son/>
- Digicert. (2003). *Capa de conexión segura SSL*. Obtenido de Digicert: <https://www.digicert.com/es/ssl.htm>
- Domínguez, Y., & Urra González, P. (12 de Septiembre de 2006). *Sistemas de gestión de contenidos: En busca de una plataforma ideal*. Obtenido de Biblioteca Virtual en Salud de Cuba: [http://bvs.sld.cu/revistas/aci/vol14\\_4\\_06/aci11406.htm#cargo](http://bvs.sld.cu/revistas/aci/vol14_4_06/aci11406.htm#cargo)
- Dorantez, C. A. (2015). *Platzi*. Obtenido de Platzi: <https://platzi.com/blog/intro-react-js/>
- Express. (2019). *Express*. Obtenido de Express: <https://expressjs.com/>
- García Oterino, A. (24 de Julio de 2015). *¿Qué es Docker? ¿Para qué se utiliza? Explicado de forma sencilla*. Obtenido de <https://www.javiergarzas.com/2015/07/que-es-docker-sencillo.html>
- García, N. A. (Julio de 2012). *amazonaws*. Obtenido de amazonaws: <http://tesis.pucp.edu.pe/repositorio/handle/123456789/1445>
- GitHub. (2019). *Apollo Client*. Obtenido de GitHub: <https://github.com/apollographql/apollo-client>
- GraphQL. (2019). *Running an Express GraphQL Server*. Obtenido de GraphQL: <https://graphql.org/graphql-js/running-an-express-graphql-server/>
- Grupo Garutu. IT Solutions. (s.f.). *DBaaS: El futuro en el éxito de nuestras empresas está en las bases de datos en la nube*. Obtenido de Grupo Garutu. IT Solutions: <https://grupogarutu.com/dbaas-bases-datos-la-nube/>
- Hernández, A. (10 de Octubre de 2014). *Sistemas de gestión de contenidos historia y su evolución*. Obtenido de Formación Digital: <https://cursodemarketingdigitalocc.es/sistemas-de-gestion-de-contenidos-historia-y-su-evolucion/>

- Jiménez, M., & Moreo, M. (2006). *Diseño de portales mediante CMS*. Obtenido de <http://lsi.ugr.es/~jparets/Escritos/Dise%F1o%20de%20portales%20median%20CMS.pdf>
- Joskowicz, J. (10 de Febrero de 2008). *Reglas y Prácticas en eXtreme Programming*. Obtenido de Universidad de la República: <https://iie.fing.edu.uy/~josej/docs/XP%20-%20Jose%20Joskowicz.pdf>
- Karnik, N. (11 de Febrero de 2018). *Introduction to Mongoose for MongoDB*. Obtenido de [freeCodeCamp.org](https://medium.freecodecamp.org/introduction-to-mongoose-for-mongodb-d2a7aa593c57): <https://medium.freecodecamp.org/introduction-to-mongoose-for-mongodb-d2a7aa593c57>
- Lázaro, D. (2016). *Ataques SQL Injection en PHP*. Obtenido de Diego Lázaro. Desarrollo web en PHP y SEO: <https://diego.com.es/ataques-sql-injection-en-php>
- Lázaro, D. (2016). *Ataques XSS: Cross-Site Scripting en PHP*. Obtenido de Diego Lázaro. Desarrollo web en PHP y SEO: <https://diego.com.es/ataques-xss-cross-site-scripting-en-php>
- Lázaro, D. (2016). *Configuración de la seguridad en PHP*. Obtenido de Diego Lázaro. Desarrollador web y SEO: <https://diego.com.es/configuracion-de-la-seguridad-en-php>
- Lázaro, D. (2016). *Escape de datos de salida en PHP*. Obtenido de Diego Lázaro. Desarrollo web en PHP y SEO: <https://diego.com.es/escape-de-datos-de-salida-en-php>
- Lázaro, D. (2016). *Filtrado de datos de entrada en PHP*. Obtenido de Diego Lázaro. Desarrollo web en PHP y SEO: <https://diego.com.es/filtrado-de-datos-de-entrada-en-php>
- Lázaro, D. (2016). *Seguridad en el almacenamiento de datos en PHP*. Obtenido de Diego Lázaro. Desarrollo web en PHP y SEO: <https://diego.com.es/seguridad-en-el-almacenamiento-de-datos-en-php>



- Lázaro, D. (2016). *SSL y OpenSSL en PHP*. Obtenido de Diego Lázaro. Desarrollo web en PHP y SEO: <https://diego.com.es/ssl-y-openssl-en-php>
- Ley No. 787 (29 de Marzo de 2012). Obtenido de <http://legislacion.asamblea.gob.ni/normaweb.nsf/9e314815a08d4a6206257265005d21f9/e5d37e9b4827fc06062579ed0076ce1d>
- Luján Mora, S. (s.f.). *Programación en Internet. Clientes WEB*. Alicante: Club Universitario.
- Michelinakis, D. (Agosto de 2004). *Open Source Content Management Systems*. The University of Warwick.
- Microsoft Azure Contributors. (2014). *How to Use the SendGrid Email Service from PHP*. Obtenido de Microsoft Azure: <https://docs.microsoft.com/en-us/azure/store-sendgrid-php-how-to-send-email>
- Mier y Terán, A., & Vasquez Martínez, M. (25 de Mayo de 2016). *Aspectos Básicos de la Seguridad en Aplicaciones Web*. Obtenido de Universidad Autónoma de México: <https://www.seguridad.unam.mx/historico/documento/index.html-id=17>
- mLab. (2019). *Database-as-a-Service for MongoDB*. Obtenido de mLab: <https://mlab.com/>
- Mozilla Developer Network. (09 de mayo de 2017). Obtenido de Mozilla Developer Network: <https://developer.mozilla.org/es/docs/Web/HTML>
- NetConsulting . (30 de 08 de 2015). *Node.js: ¿Qué es y para que sirve NodeJS?* Obtenido de NetConsulting: <https://www.netconsulting.es/blog/nodejs/>
- Pérez, M., & Gutiérrez, M. (Junio de 2005). *Sistemas de gestión de contenidos en la gestión del conocimiento*. Obtenido de Textos universitarios de biblioteconomía i documentació: <http://bid.ub.edu/14monto2.htm>
- Pitt, C. (2012). *Pro-PHP MVC*. New York: Springer Science+Business Media New York.

- Platzi. (2017). *¿Qué es Heroku y para qué me sirve?* Obtenido de Platzi: <https://platzi.com/blog/que-es-heroku-y-para-que-me-sirve/>
- PMOinformatica. (Junio de 2014). *Plantilla de casos de prueba*. Obtenido de PMOinformatica.com: <http://www.pmoinformatica.com/2014/06/plantilla-de-casos-de-prueba.html>
- PostgreSQL. (02 de Octubre de 2010). Obtenido de PostgreSQL: [http://www.postgresql.org.es/sobre\\_postgresql](http://www.postgresql.org.es/sobre_postgresql)
- PowerData. (8 de Septiembre de 2014). *Qué es la arquitectura orientada a servicios SOA*. Obtenido de PowerData: <https://blog.powerdata.es/el-valor-de-la-gestion-de-datos/bid/394442/qu-es-la-arquitectura-orientada-a-servicios-soa>
- Pressman, R. S. (2010). *Software engineering : a practitioner's approach*. New York: McGraw-Hil.
- RedHat. (2019). *¿Qué es Docker?* Obtenido de RedHat: <https://www.redhat.com/es/topics/containers/what-is-docker>
- Reyes, J. (24 de Noviembre de 2009). *How to Evaluate What CMS to Use*. Obtenido de SIX REVISIONS: <http://sixrevisions.com/web-development/how-to-evaluate-what-cms-to-use/>
- Roel, A. (s.f.). *El protocolo SSL*. Obtenido de Facultad de Ciencias Exactas - UNLP: <http://penta2.ufrgs.br/gereseg/unlp/tut1998/ssl.htm>
- Rosell León, Y. (2011). *Sistemas gestores de contenidos: una mirada desde las ciencias de la información*. Obtenido de ACIMED: [http://scielo.sld.cu/scielo.php?script=sci\\_arttext&pid=S1024-94352011000100002&lng=es&tlng=es](http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1024-94352011000100002&lng=es&tlng=es)
- Santa María, L. (25 de Junio de 2013). *Todo lo que necesitas saber sobre los Sistemas de Gestión de Contenido – Historia de los CMS*. Obtenido de StaffCreativa: <http://www.staffcreativa.pe/blog/historia-cms/>

- Saurabh, P. (1 de Abril de 2018). *Continuous Integration using Travis on Github*. Obtenido de Hackernoon: <https://hackernoon.com/continuous-integration-using-travis-on-github-1f7f2314b6b7>
- Suter, R. (2012). *MongoDB. An introduction and performance analysis*. Obtenido de <https://wiki.hsr.ch/Datenbanken/files/MongoDB.pdf>
- Taylor, A. (7 de Septiembre de 2018). *Crafting Beautiful UIs in React Using Ant Design*. Obtenido de Alligator.io: <https://alligator.io/react/beautiful-uis-ant-design/>
- The Open Group. (2019). *Service-Oriented Architecture – What Is SOA?* Obtenido de The Open Group: [http://www.opengroup.org/soa/source-book/soa/p1.htm#soa\\_definition](http://www.opengroup.org/soa/source-book/soa/p1.htm#soa_definition)
- TIC Portal. (2019). *Amazon Web Services*. Obtenido de TIC Portal: <https://www.ticportal.es/temas/cloud-computing/amazon-web-services>
- Universidad Unión Bolivariana. (2008). *PROGRAMACIÓN EXTREMA (XP)*. Obtenido de Universidad Unión Bolivariana: <http://ingenieriadesoftware.mex.tl/images/18149/PROGRAMACI%C3%93N%20EXTREMA.pdf>
- Urra González, P., & Sarduy Domínguez, Y. (2006). *Sistemas de gestión de contenidos: En busca de una plataforma ideal. ACIMED*. Obtenido de ACIMED.
- Urrutia, V. (22 de Diciembre de 2018). *Qué es Heroku, para qué sirve, ventajas y desventajas*. Obtenido de VIDELCLOUD. Descubriendo el hilo negro: <https://videlcloud.wordpress.com/2018/12/22/que-es-heroku-para-que-sirve-ventajas-y-desventajas/>
- Valencia, U. P. (12 de noviembre de 2011). *Ingeniería de Software y sistemas de información*. Obtenido de <http://issi.dsic.upv.es/archives/f-1069167248521/actas.pdf>

- Vargas Cordero, Z. R. (8 de Agosto de 2009). *La investigación aplicada: Una forma de conocer las realidades con evidencia científica*. Obtenido de redalyc: <http://www.redalyc.org/html/440/44015082010/>
- Vázquez Villalta, J., & Sotelo Almendarez, R. (2014). *Sistema Gestor de Contenido-Indexación e HibernateSearch (SiGeC-INIBAS) (Para optar al título de Ingeniero en Computación)*. Managua: Universidad Nacional de Ingeniería.
- Viciano Pérez, A. (2014). *Venta Online*. ic Editorial.
- W3. (2017). *STANDARDS*. Obtenido de W3C: <https://www.w3.org/standards/>
- Yanes Enriquez, O., & Gracia del Busto, H. (2011). Mapeo Objeto / Relacional (ORM). *Telemática*, 1-7.
- Zepeda, A. A. (2017). *Desarrollo del sistema web para la administración de contenido, gestión de contratos de suscripciones de la empresa iNOVATE*. Managua.
- Ziroll, B. (7 de Diciembre de 2015). *Mongoose Schema Basics*. Obtenido de V School | Human-centered Ed Tech: <https://coursework.vschool.io/mongoose-schemas/>

# Anexos

## **Glosario**

**Establecimientos clientes:** Son todas aquellas empresas clientes directos de Wimix Solutions and Services, las cuales hacen uso de sus servicios y del sistema actual.

**Usuario Wimix:** cualquier agente trabajador de Wimix Solutions and Services que posee acceso al sistema y cuenta con todos los permisos para hacer uso del sistema.

**Usuario general:** agente trabajador de cualquier establecimiento cliente que cuente con una cuenta de usuario en el sistema, la cual cuenta con privilegios básicos para el uso del sistema.

## **Manual de Usuario**

### **I. Inicio de sesión**

La interfaz de inicio de sesión cuenta con dos campos básicos donde usted, como usuario, deberá ingresar su nombre de usuario y su contraseña.

1. Ingrese su nombre de usuario y contraseña.
2. Presione el botón “Entrar” para iniciar sesión



En caso de que el sistema arroje un mensaje de error, por favor contactarse con su supervisor o un agente de Wimix para restablecimiento de contraseña o creación de cuenta.

### **II. Formulario de registro**

1. En la sección de formulario de registro, podrá seleccionar una de las opciones de plantillas propuestas.
2. Una vez seleccionada una, aparecerá una nueva ventana dando la opción de Guardar.
3. De clic sobre el botón de Guardar una vez esté seguro de su elección.

The screenshot displays the 'Formularios' (Forms) management interface. At the top, there are two tabs: 'Formularios' (active) and 'Historial'. Below the tabs is a grid of five template cards. Each card shows the template name and the number of fields it contains. The templates are: Plantilla 1 (4 fields), Plantilla 2 (2 fields), Plantilla 3 (3 fields), Plantilla 4 (5 fields), and Plantilla 5 (2 fields). To the right of the grid is a 'Nueva plantilla' (New template) button with a 'Limite de Campos: 4' (Field limit: 4) label. The footer of the interface reads 'Wimix Solutions and Service S.A. ©2019'.

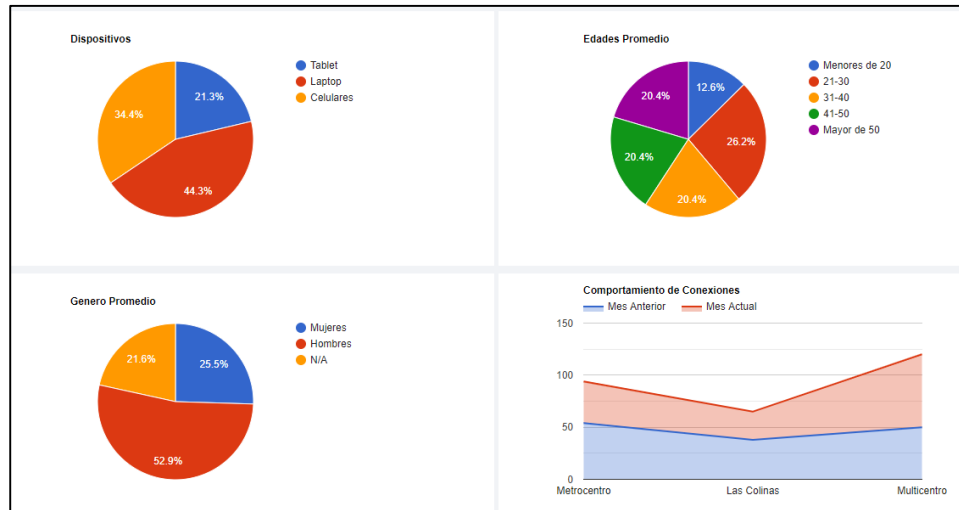
**En caso de querer crear una nueva plantilla:**

1. Seleccionar la opción: Nueva Plantilla.
2. Seleccionar las distintas opciones que desea para su formulario. Puede elegir 4 de las 5 opciones proporcionadas:
  - a. Nombre
  - b. Correo
  - c. Fecha de Nacimiento
  - d. Teléfono
  - e. Género
3. Dar clic sobre el botón de Guardar
4. Se abrirá una nueva ventana mostrando una pre visualización de su formulario creado.
5. Dar clic sobre el botón Guardar para aceptar los campos.



### III. Estadísticas

La sección Estadísticas muestra un gráfico general de los distintos dispositivos conectados a lo largo del tiempo.



### IV. Usuarios

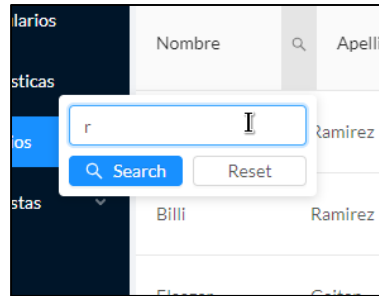
Para llegar a la sección de Usuarios, puede hacerlo desde el menú lateral del sistema.

La sección Usuarios muestra todos los usuarios registrados desde el portal cautivo. Genera una tabla de los datos de registro solicitados en Formularios de Registro.

| Nombre | Apellido | Correo Electronico | Genero | Telefono | Fecha de Ingreso | Send Email    |
|--------|----------|--------------------|--------|----------|------------------|---------------|
|        |          |                    |        |          |                  | Enviar Correo |
|        |          |                    |        |          |                  | Enviar Correo |
|        |          |                    |        |          |                  | Enviar Correo |

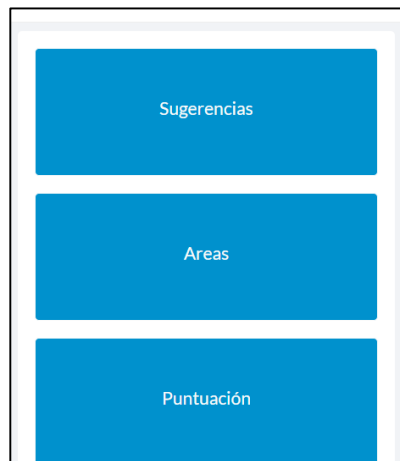
#### Filtro de datos

Para filtrar los datos simplemente haga clic en el botón de búsqueda alojado a la derecha de cada nombre de la columna. Automáticamente el sistema buscará los registros solicitados.



## V. Encuestas NPS

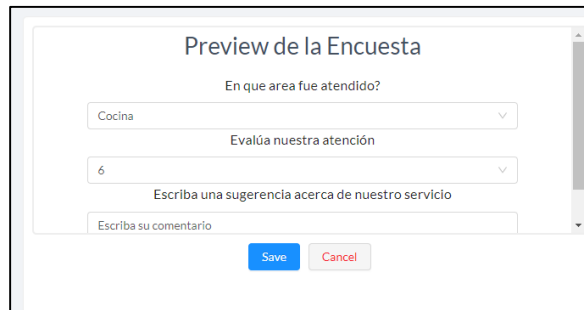
La sección de encuestas NPS se encuentra en el menú lateral > **Encuestas**  
> **NPS**



1. Para llenar una encuesta NPS deberá rellenar las 3 opciones que aparecen: área, sugerencia y puntuación.
  - Opción **Área**
    1. Agregue una nueva etiqueta dando clic en el botón **Nueva etiqueta**
    2. Una vez satisfecho con su selección, de clic en el botón **Guardar**.
  - Opción **Sugerencia**
    1. Agregue al campo de texto la pregunta que desea que sus usuarios respondan.
    2. Una vez satisfecho con su selección, de clic en el botón **Guardar**
  - Opción **Puntuación**
    1. Escriba en el campo de texto, la descripción sobre lo que se quiere evaluar. Por ejemplo: “¿Cómo calificaría nuestros servicios?”

2. Automáticamente el sistema genera una lista de puntuación válida para que el usuario pueda seleccionar.
3. De clic en el botón **Guardar**

Una vez completo todo el proceso de relleno de las 3 opciones. El sistema mostrará una pre visualización de su encuesta. Si se encuentra satisfecho con los resultados, de clic en el botón **Guardar**



Preview de la Encuesta

En que area fue atendido?

Cocina

Evalúa nuestra atención

6

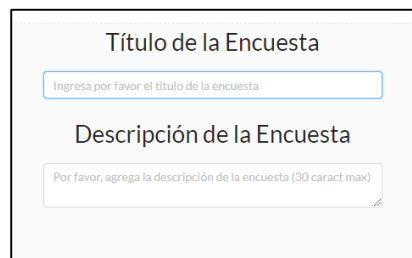
Escriba una sugerencia acerca de nuestro servicio

Escriba su comentario

Save Cancel

## VI. Encuestas Personalizadas

La sección de encuestas personalizadas se encuentra en el menú lateral > **Encuestas > Personalizadas**



Título de la Encuesta

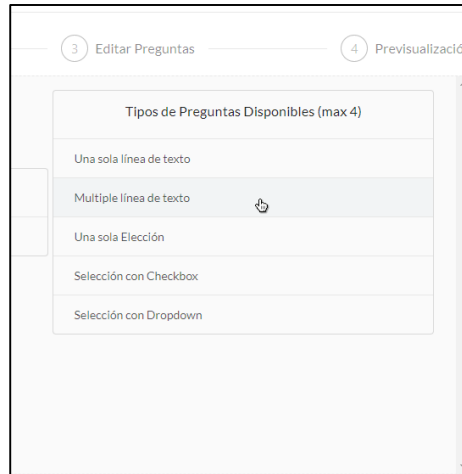
Ingresa por favor el título de la encuesta

Descripción de la Encuesta

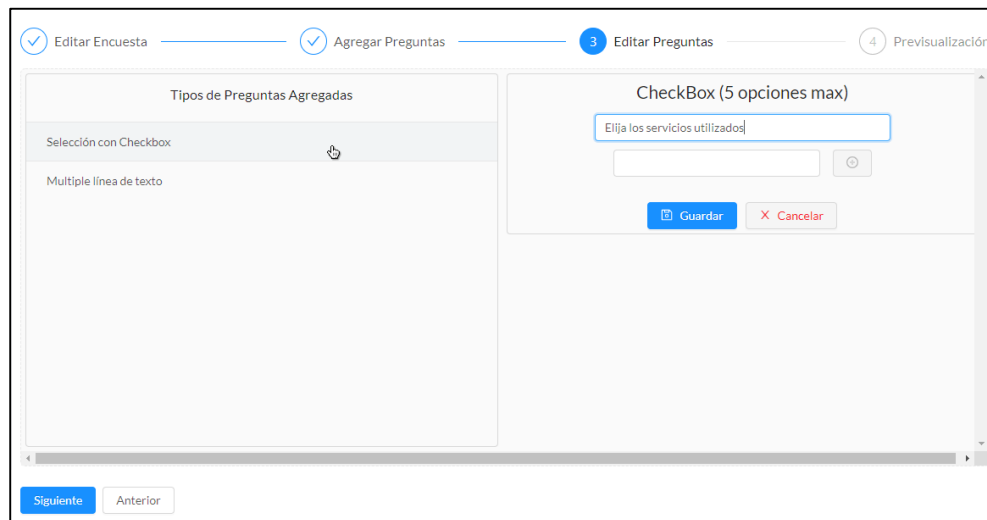
Por favor, agrega la descripción de la encuesta (30 caract max)

Para diseñar una encuesta personalizada puede ejecutar los siguientes pasos:

1. Escriba un título y descripción para la encuesta, ambos deben describir de manera resumida, el objetivo de la encuesta.
2. Procede a dar clic en el botón **Siguiente**.
3. Para agregar preguntas a una encuesta personalizada, deberá seleccionar 4 de las opciones de la lista a la derecha de la pantalla, haciendo clic en ellas.



4. Una vez seleccionadas las opciones que prefiera, da clic en el botón de **Siguiente**
5. Seguidamente se procede a editar las preguntas seleccionándolas del panel izquierdo. Y luego escribiendo de acuerdo a las preguntas consideradas necesarias.
6. Se procede a dar clic en el botón **Siguiente**.

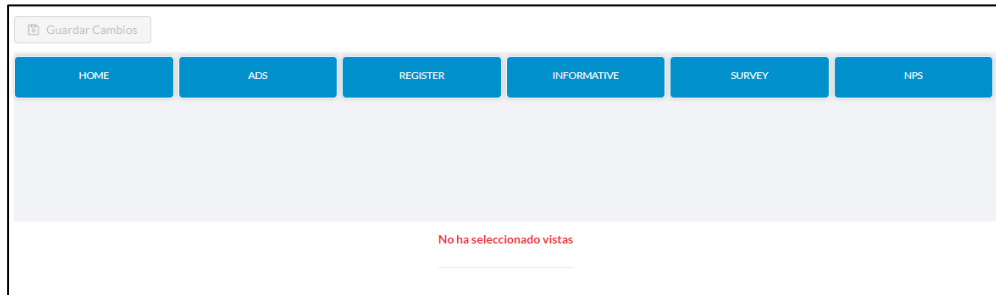


7. Finalmente, el sistema mostrará una pre visualización de su encuesta. Si se encuentra satisfecho con los resultados, de clic en el botón **Guardar**.

## **VII. Pasos**

Puede ingresar a la sección Pasos desde el menú lateral.

1. Para editar el orden de los pasos del portal cautivo, deberá seleccionar 3 de los 6 pasos que aparecen como pasos.



2. Una vez seleccionado los 3 pasos, puede reordenarlos arrastrándolos según el orden deseado o eliminarlos.

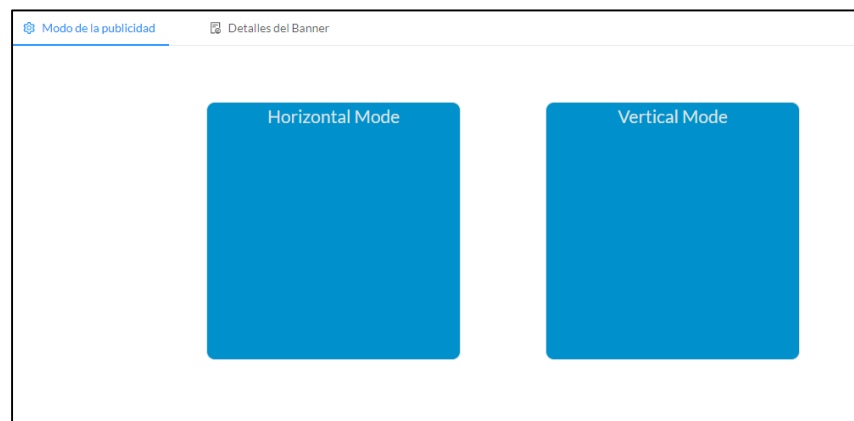


3. Finalmente, puede dar clic al botón **Guardar**

## **VIII. Publicidad**

En esta sección puede configurar la forma en que sus clientes pueden ver la publicidad en el portal cautivo. La sección de publicidad puede encontrarla en el menú lateral.

1. Para seleccionar una opción simplemente debe dar clic a la deseada y el sistema guardará automáticamente.

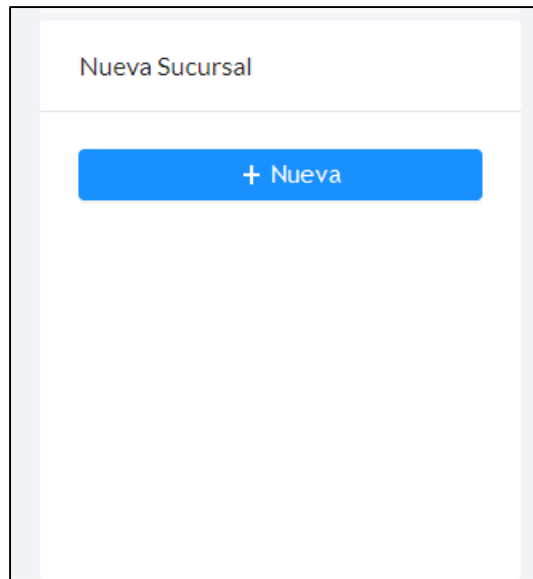


## IX. Configuración

La sección de Configuración le permitirá crear sucursales, ver usuarios del sistema y configurar otros aspectos de la aplicación. La sección de Configuración la encontrará en el menú lateral.

### Configurar una sucursal

1. Para configurar una sucursal, deberá dar clic en el botón **Nueva**



2. Aparecerá una ventana pop-up en donde deberá rellenar todos los campos

